Week 6 Review

Submit Assignment

Due Feb 23 by 11:59pm **Points** 20 **Submitting** a text entry box or a file upload

Overview

When attempting these problems, please keep in my mind the nature of **Academic Honesty** in this course. This week you should submit your response to the Week 6 Review. This will support your review of the topics covered this week and prepare you for writing your program.

Assessing your Problem Solutions

Recall, your work involves more than just correctness. Specifically, we will look for:

- appropriate separation of concerns (including classes)
- appropriate use of function interfaces (i.e., using parameters and return values, not using variables declared as part of a script to make them accessible in all function scopes)
- comments explaining the why behind more difficult areas of code
- documentation of classes/modules for explaining their purpose and usage
- good function, parameter, and variable names that communicate the purpose and make the program more readable

Please keep in mind there is no single correct answer for *how* you design the program: there are many good approaches that are worth full credit.

1. **[unique_list.py]** A unique list has no repeated items. For example, the list ['orange', 'juice', 'futures'] is unique because each item only appears one time. However, the list ['sell','mortimer','sell'] is not unique because the string 'sell' appears more than one time.

Write a program named `unique_list` where a user inputs a series of positive integers and, when finished, it reports whether or not the sequence of numbers input is unique or not. For example:

```
This program tests if the sequence of positive numbers you input are unique
Enter -1 to quit
Enter the first number: 9
Next: 5
Next: 3
Next: 6
Next: 23
```

```
Next: -1
The sequence [9, 5, 3, 6, 23] is unique!
```

```
This program tests if the sequence of positive numbers you input are unique
Enter -1 to quit
Enter the first number: 83
Next: 20
Next: 593
Next: 28
Next: 20
Next: 51
Next: -1
The sequence [83, 20, 593, 28, 20, 51] is NOT unique!
```

2. **[vending_status_checker.py]** Modify the vending status checker to now also report on the status of the machines overall. Be sure to click the "vending status checker" link above and save the .json files for use in the assignment.

Our current vending status checker produces an inventory report. For each machine's inventory file it is counting how many of each beverage were stocked last time, how many of that beverage are currently in stock, and how many slots the beverage occupies (i.e., each slot holds 8 beverages). Your goal is to modify the program such that it can also produce a machines' sales report, something like this:

```
Label           Pct Sold   Sales Total
REID_1F            52.40% $  230.25
REID_2F            42.02% $  138.25
REID_3F            52.07% $  214.75
```

This report contains three fields: the label of the vending machine, what percentages of the beverages it was last stocked with are sold, and how many total dollars of sales has this generated.

You will need to create a new dictionary where the keys are the vending machine labels, and the values are a new type of object called a `MachineStatus`. For each instance, the `MachineStatus` class should store:

- the label of a vending machine
- the total amount of beverages the vending machine was previously stocked with
- the total amount of beverages currently in stock in the vending machine
- the total income of the machine from the last time it was stocked until now (note: beverages have different prices, so you cannot simply multiply the change in stock times $1.50 to get the total income)

Three major modifciations to `vending_status_checker.py` are necessary:

- adding the `MachineStatus` class
- adding to the loop for counting beverages additional code for counting and updating a vending machine's `MachineStatus` object values
- enabling a user to choose the machines' sales report, not just the inventory report, and then display the report as shown in the following example

```
Would you like the (m)achine report or the (i)nventory report? m
Label           Pct Sold   Sales Total
REID_1F            52.40% $  230.25
REID_2F            42.02% $  138.25
REID_3F            52.07% $  214.75
```

HINT: Start by making sure you understand how the counting is done for inventory, most importantly how the dictionary is used to go from the name a beverage to its `InventoryItem` object. Then create a new dictionary with vending machine labels as keys that maps to the vending machine's `MachineStatus` object. There should be only one `MachineStatus` object for each vending machine.

Submission

Please post all necessary .py files to Canvas and include your answers to the questions under the "Canvas Submission" banner in the textbox provided.

Canvas Submission

When you submit this assignment here in Canvas, I would like you to answer the following question(s):

1. How many hours do you estimate you used completing this assignment?
2. What was easiest for you when completing this assignment?
3. What was the most difficult challenge you experienced when completing this assignment?

Week 6 Review Rubric

Criteria	Ratings		Pts
1. Unique List: Inputs	**2.0 pts** **Full Marks**	**0.0 pts** **No Marks**	2.0 pts
1. Unique List: Correctly identifies Uniqueness	**4.0 pts** **Full Marks**	**0.0 pts** **No Marks**	4.0 pts
1. Unique List: Output	**1.0 pts** **Full Marks**	**0.0 pts** **No Marks**	1.0 pts
2. Vending Status: Structures Machine Level Data	**2.0 pts** **Full Marks**	**0.0 pts** **No Marks**	2.0 pts
2. Vending Status: Correctly Counts Machine Data	**4.0 pts** **Full Marks**	**0.0 pts** **No Marks**	4.0 pts
2. Vending Status; Menu Choice	**1.0 pts** **Full Marks**	**0.0 pts** **No Marks**	1.0 pts
2. Vending Status: Tabular Formatted Output	**2.0 pts** **Full Marks**	**0.0 pts** **No Marks**	2.0 pts
ALL: Concerns appropriately separated	**1.0 pts** **Full Marks**	**0.0 pts** **No Marks**	1.0 pts
ALL: Commenting/Documentation present in more difficult areas	**1.0 pts** **Full Marks**	**0.0 pts** **No Marks**	1.0 pts
ALL: Descriptive, readable, informative identifiers for classes, variables, and functions	**1.0 pts** **Full Marks**	**0.0 pts** **No Marks**	1.0 pts
Answers Questions in Canvas Submission	**1.0 pts** **Full Marks**	**0.0 pts** **No Marks**	1.0 pts
			Total Points: 20.0