Exam guidelines

Please type all code yourself from your knowledge. You may use any standard reference we’ve used in class.

BUS 175 20S
Exam 2: Yet Another Receipt Generator (YARG)

Goal

Use any material covered up to and including Week 8: Dictionaries to write a more involved receipt generator.

Background

Your hard work has not gone unnoticed at Sephulta, the personal care and beauty store where you work. The management team would like you to write a point-of-sale program that can log purchases and generate a receipt for the customer. Some items are discounted at various rates, and the final total must reflect discounts. The final total must also include 15% tax (the cost of living in Southern California is quite high).

Setup

Copy the starter code found here:
https://repl.it/@richyueh/BUS-175-20S-final-exam-starter-code
Paste the code into the first two lines of your program and keep them there. Be sure not to change anything. This code generates an empty list called `items` and a dictionary called `discounts` that contains discount keywords and their discount rates as a decimal.

What your Python program should do

Ask the user to enter an item to buy. The item could be anything. For each item, ask for its name, its cost, and the amount of that item the customer wants to buy. Once you have the three pieces of information, store them into the `items` list.

Ask the user if they want to continue adding items and give them the option of "y" for yes and "n" for no. If they choose "y," repeat the process above.

Once the user chooses not to continue adding items, your program should do two things. First, print to the screen an actual list of each item purchased, the amount of each item, and the total cost for that item including discounts. Also show the subtotal for all items and a final total including tax. Second, print this same information to a `receipt.txt` file.

If the item name contains a keyword found in discounts, apply the associated discount rate to the item's subtotal.

Tax is 15% and applied to the subtotal of all items. The final total should include tax.

Important: required functions!

Each task of the program must be in a separate function. Your program must contain the following functions, defined exactly as they are listed below and including the parameters shown. You are not allowed to use the `global` keyword in Python.

```
def calculate_tax(total)

def calculate_discount(items)

def calculate_total(items)

def if_discount(items)    # Checks if any item in the purchase
list qualifies for a discount.

def print_receipt(items)

def main()                # Please put your non-function code
into a main() function.
```

You should not need to define any additional functions.

Walkthrough output

```
Item to buy: red lipstick
Item cost: 10
Amount to buy: 1
Buy another item? (y/n): n
red lipstick        1:9.50
--------------------------------------------------
Subtotal: 9.50
Total after tax: 10.92
```

This item contains the discount keyword “lipstick” so a discount of 5% is applied. The 5% came from the `discounts` dictionary.

How to read this: One item, total price 9.50.

Tax is 15%. Minor rounding differences at the hundredth position (.0x) are OK.

```
red lipstick       1:9.50
--------------------------------------------------
Subtotal: 9.50
Total after tax: 10.92
```

This is what `receipt.txt` looks like after the first run. You can set up your receipt function to print the exact same thing to the screen and to the file.

```
Item to buy: perfume annual sale
Item cost: 100
Amount to buy: 1
Buy another item? (y/n): y
Item to buy: eye skincare
Item cost: 10
Amount to buy: 1
Buy another item? (y/n): n
perfume annual sale      1:80.00
eye skincare        1:9.70
--------------------------------------------------
Subtotal: 89.70
Total after tax: 103.16
```

The two words “annual sale” are a discount keyword. Your program must be able to recognize this.

If two or more discount keywords appear in an item, take the first keyword encountered in the dictionary order. Here you can see the item is discounted by 3%, and 3% comes from the “eye” keyword discount. The 20% “skincare” discount is not applied.

Additional sample output can be found at the bottom of this document.

Assumptions and clarifications

Please join this new Campuswire group to discuss any questions about what can or should happen: https://campuswire.com/p/G7325CC04 (If asked, the invite code is 6484.)

- For simplicity, you do not need to worry about formatting or making things pretty. You do not need dollar signs, verbose text, etc. You are welcome to include formatting if you have time.
- For simplicity, you do not need to worry about input validation. Assume the user will type in a valid item cost, item amount, and only valid choices for y / n.
- Your program must handle decimals in the appropriate places.
- The item name can literally be anything. See the sample output below.
- You are not allowed to use the `global` keyword in Python. If you really, truly cannot get it to work without using `global`, go ahead and use `global` (you might need to do some research on Google). You will lose some points for `global`.
- Clarifying the assumption above and the required functions, this means that you must pass the `items` list to each function that uses it. This is why you see `(items)` in the function definitions. For the `calculate_tax` function, you must pass in a subtotal amount.
- Hint: Some of the functions may need to return a value.
- Hint: Contrary to what Prof. Rich said in the final review Zoom, lists *will* play an important part in this final exam. You will be using material from each main week of class.

How to work on and submit your Python code

Please follow the same instructions with repl as you have done throughout the class.

Deliverables

Submit your repl share link to iLearn *before the due date and time* listed on the iLearn submission link.

Grading Rubric

200 points total.

The important grading points are properly setting up the required functions and applying discounts to items. If you get the functions working and can correctly apply discounts, that's the majority of your program and thus any other grading points are less important.

Additional sample output

```
Item to buy: body wash
Item cost: 27.99
Amount to buy: 2
Buy another item? (y/n): y
Item to buy: item name blah blah
Item cost: 500
Amount to buy: 5
Buy another item? (y/n): y
Item to buy: lipstick eye skincare
Item cost: 100
Amount to buy: 1
Buy another item? (y/n): n
body wash      2:55.98
item name blah blah      5:2500.00
lipstick eye skincare         1:95.00
--------------------------------------------------
Subtotal: 2650.98
Total after tax: 3048.63
```