

**ANL252 (Online)**

**Python for Data Analytics**

# **Tutor-Marked Assignment**

**July 2023 Presentation**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **Don Lee Yang** | **W2270423** |

**Instructor’s Name: Mr. MUNISH KUMAR**

**Submission Date: 13/09/2023**

**Question 1a**

Source code plagiarism is copying another programmer's source code and claiming it as their own.

There are a variety of reasons why source code plagiarism happens. Here are just a few reasons why it occurs:

* **Online accessibility**: Coding from home allows students to access online coding libraries and free-to-use codes (Carmil, 2022).
* **Academic Pressure**: Students might struggle and look for shortcuts to complete assignments (Lee, 2023).
* **Stress**: Students might face deadlines and copy codes directly online.

To avoid source code plagiarism from happening, students can do the following:

* **Avoid procrastination**: Do not leave assignments till the last minute.
* **Practice**: Take the time to keep practicing codes to understand how specific code works correctly.
* **Use codes as references**: Instead of copying codes, students can use them as referrals and credit the source code afterward.
* **Plagiarism checker**: Student can run their codes on plagiarism checker websites such as GitHub and CopyLeaks.

  In conclusion, students should do their best to ensure their work is original and always take the time to go through several ways to avoid source code plagiarism.

(175 words, excluding references)

**Question 1b**

#Inserting new items in an inventory (items and cost)

1. print ("All prices are priced in /kg \n")

2. CurMeatInv = {"chicken": 7.0, "beef": 10.0, "mutton": 9.3}

3. CurVegInv = {"chye sim": 2.0, "kang kong": 2.50, "spinach": 2.30 }

4. print ("1 for Display to Price for items \n2 to Add new Item and Price\n")

5. insertNo = int(input("Please choose your option:"))

#To Display the desired inventory

6. if insertNo == 1:

7. disCat = str(input("Choose your Category (Meat, Vegetables):"))

8. if chooseCat == "Meat":

9. for m1 , p1 in list(CurMeatInv.items()):

10. print (m1, ": $", p1)

11. elif chooseCat == "Vegetables":

12. for v1 , p2 in list(CurVegInv.items()):

13. print (v1 , ": $", p2)

#To add new items into the inventory

14. if insertNo == 2:

15. chooseCat = str(input("Which Category would you like to input the items in? (Meat, Vegetables):"))

16. if chooseCat == "Meat":

17. newMeat = str(input("Input Item Name:"))

18. newMPrice = float(input("Input Item Price:"))

19. CurMeatInv [newMeat] = newMPrice

20. print ("\nUpdated Meat Inventory")

21. for m1 , p1 in list(CurMeatInv.items()):

22. print (m1, ": $", p1)

23. elif chooseCat == "Vegetables":

24. newVeg = str(input("Input Item Name:"))

25. newVPrice = float(input("Input Item Price:"))

26. CurVegInv [newVeg] = newVPrice

27. print ("\nUpdated Vegetables Inventory")

28. for m1 , p1 in list(CurVegInv.items()):

29. print (m1, ": $", p1)

Total code lines: 29

The code above allows the user to display and insert new items into two categories of groceries, namely, meat and vegetables.

Between lines 1 and 3, two dictionaries were created for vegetables and meat. The two dictionaries consist of a key, which is the items in the dictionaries, and the value of each key, which is the price of each item.

Between lines 4 and 5, the user will be prompted to insert “1” or “2”. From line 6 to line 13, when the user inserts “1”, the user will be prompted to choose between “Meat” or “Vegetables.” When either option is selected, the preferred option will be displayed.

Between lines 13 and 29, the user can add new items and prices to both inventories. Users will be prompted to choose which category to add the item in. Then, they will be prompted to add the item and price to the inventory. The updated inventory will then be displayed.

(159 Words)

**Question 1c**

#inventory system

print ("Welcome to my inventory management system on python! \n")

print ("Insert '1' to Display items and price.")

print ("Insert '2' to Add new item and price.")

print ("Do note that all prices indicated are in per kg.")

meatInventory = {"chicken": 7.0, "beef": 10.0, "mutton": 9.3}

vegInventory = {"chye sim": 2.0, "kang kong": 2.50, "spinach": 2.30 }

category = str(input("Please insert '1' or '2' as your option:"))

#To Display the desired inventory

if category == '1':

disCat = str(input("Which category would you like to display, meat or vegetables?"))

if disCat == "meat":

for m1 , p1 in list(meatInventory.items()):

print (m1, ": $", p1)

elif disCat == "vegetables":

for v1 , p2 in list(vegInventory.items()):

print (v1 , ": $", p2)

#To add new items into the inventory

elif category == '2':

chooseCat = str(input("Which Category would you like to input the items in, meat or vegetables? "))

if chooseCat == "meat":

newMeat = str(input("Input Item Name:"))

price = float(input("Input Item Price:"))

meatInventory [newMeat] = price

print (f'{newMeat} has been added to the inventory for ${price}!')

elif chooseCat == "vegetables":

newVegetable = str(input("Input Item Name:"))

price = float(input("Input Item Price:"))

vegInventory [newVegetable] = price

print (f'{newVegetable} has been added to the inventory for ${price}!')

#Neither options were chosen.

else:

print ("Please choose either 'meat' or 'vegetables', Thank you!”)

To prevent plagiarism, I have re-written the code in question 1b and changed several line codes for various reasons. The reasons listed are as follows:

* Uniformity: To ensure uniformity throughout the code, numerous variables were renamed.
* Readability: Through several additional code, users are able to read instructions more clearly than the previous code.
* User guidance: Code lines were introduced to help users input the correct settings and to help prevent errors.

In conclusion, codes were added and re-written to improve the consistency, menu for users and ensuring the code is user-friendly, still maintaining the same functionality.

(96 Words)

**Question 2**

products = ['laptop', 'mouse', 'webcam', 'keyboard', 'speaker']

query = 'yes'

cart\_shopping = []

print ("The products available are listed below: \n")

for i in products:

print (i)

while query == 'yes':

item = str(input("Hello! Enter your the product you want: ")).lower()

if item not in products:

print ("The product you entered is not available, try again!")

continue

price\_of\_item = float(input ("How much is it? ($) "))

add\_cart = [item, price\_of\_item ]

cart\_shopping.append (add\_cart)

query = str(input ('Would you like to continue? (yes/no): ')).lower

print ("You have added the following: ")

for item, price in (cart\_shopping):

print(item, ': ', '$', price )

Readability: Modify the code where users are able to read the product clearer, making it more presentable.

Structured variable: Change variable names to allow other coders to better understand each variable.

Case sensitivity: Added “.lower()” to allow prevent any syntax error from occurring.

Output**:** Modified several lines of code to allow the printed output to be clearer for the user to understand.

**Reference**

1. Carmil. (2022). All you need to know about plagiarism in coding. *Copyleaks*  
https://copyleaks.com/blog/all-you-need-to-know-about-plagiarism-in-coding

2. Lee, C. (2023, September 5). What is source code plagiarism and what does it have to do with academic integrity? *Turnitin*. https://www.turnitin.com/blog/what-is-source-code-plagiarism-and-what-does-it-have-to-do-with-academic-integrity