

**ANL252**

**Python for Data Analytics**

# **Tutor-Marked Assignment**

**July 2023 Presentation**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **Naomi Tina Gan Min** | **Z2210445** |
|  |  |
|  |  |
|  |  |

**Tutorial Group: ­­­­­­­­­­­T 03**

**Instructor’s Name: Mr Kumar Munish**

**Submission Date: 15/09/2023**

**Question 1a**

Plagarism in coding happens when a code is copied without credit or valid authorisation, and happens most often in academic assessments. Due to a lack of understanding, inexperienced coders may not even be aware of their offences. (Copyleaks, 2021)

Direct Copy-Paste: Developers may copy code from online sources or colleagues without attribution, passing it off as their own.

Code Repositories: Cloning or forking repositories and presenting them as original work without modifications.

Code Rebranding: Slightly altering variable names or comments in plagiarized code to make it seem different.

Inadequate Documentation: Failing to document borrowed code or attributing it correctly within the codebase or documentation.

There are three main solutions I have discovered to prevent coding plagarism from reoccuring:

Open Source: Leverage open-source libraries with proper licensing.

Use Documentation: Always reference and cite external code sources.

Code Review: Regularly review and check code for plagiarism.

While most computer codes contain ideas borrowed from somewhere else, it is important to properly understand the resources utilised and promote ethical coding practices.

**[Word Count: 168]**

**Question 1b**

people = 30

cars = 40

trucks = 15

if cars > people:

print("We should take the cars.")

elif cars < people:

print("We should not take the cars.")

else:

print("We can't decide.")

if trucks > cars:

print("That's too many trucks.")

elif trucks < cars:

print("Maybe we should take the trucks.")

else:

print("We still can't decide.")

if people > trucks:

print("Alright, let's just take the trucks.")

else:

print("Fine, let's just stay home then.")

(Taken from Learn Python 3 the Hard Way: A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code, Zed A. Shaw (Pearson), Retrieved 2023)

Lines 1-3 indicdate the number of people, cars and trucks respectively.

Lines 6-11 take into consideration the use of cars. If the number of cars are more than people, they will take the cars, if not they will not take the cars. The else function would pop up if the number of people = to the number of cars. Same rationale applies to lines 13-18 when they compare trucks to cars. Finally, lines 20-23 only consider the number of people to number of trucks. If there are more people, they will go via trucks, if not they will stay home.

**[Word Count: 198]**

**Question 1c**

people = int(input("How many people are in your group?"))

cars = int(input("How many cars do you have?"))

trucks = int(input("How many trucks do you have?"))

if cars > people and trucks > people:

print("We can choose between cars or trucks.")

elif cars > people or cars == people:

print("We should take the cars.")

else:

print("We can't decide.")

if trucks == people:

print("Maybe we should take the trucks.")

else:

print("We still can't decide.")

if cars == trucks:

print("Should we split up and take both cars and trucks?")

else:

print("Fine, let's just stay home then.")

Output:

How many people are in your group? 15(ans)

How many cars do you have? 20(ans)

How many trucks do you have? 20(ans)

We can choose between cars or trucks.

We still can't decide.

Should we split up and take both cars and trucks?

Change 1: Allows people to input the numbers. This code can be used by non-coders because they can enter the data and then have the options provided accordingly.

Change 2: Choice between trucks and cars if they are both more than the number of people. In 1b, users had the choice made according to the number of cars outweighing the number of people, followed by the number of trucks outweighing the cars and finally the number of trucks outweighing the number of people. I found this ineffective in presenting possible solutions as cars were automatically eliminated from the option if there were less than people.

Change 3: Having the same number of people to trucks or cars still allows them to choose either vehicle, instead of being considered ‘unable to decide’.

**[Word Count: 275]**

**Question 2**

The current code in appendix 1 is quite disorganised. For usability, lines 5-10 have been changed to allow continuos items to be added to cart, with the option to exit by entering “d”. Lines 11-14 allow prices to be input in the rows below the products. For readability, the prices have been changed from (in SGD) to S$ with a float(input). Finally for overall quality, a summary of their cart items have been added in lines 16-25. The output will be a list of products and the sum of all prices.

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

prices = []

total = 0

print(f'We have a list of products here: {products}.')

while True:

product = input("Hello! What do you want to buy, (e to exit): ?")

if item not in products:

print(f'Wrong product! Please try again.')

elif product == "d":

break

else:

price = float(input(f"Enter the price of a {product}: S$"))

products.append(product)

prices.append(price)

print("This is our updated shopping list")

for product in products:

print(product)

for price in prices:

total = total + price

print()

print(f"Your total is: S${total}")

**[Word Count: 185]**

**Reference List**

C. (2021, July 9). *All You Need to Know About Plagiarism in Coding | Copyleaks*. Copyleaks. <https://copyleaks.com/blog/all-you-need-to-know-about-plagiarism-in-coding>

Shaw. (19 C.E., September 13). *Learn Python 3 the Hard Way: A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code* (1st ed.).