w

**ANL252**

**Python For Data Analytics**

# **Tutor-Marked Assignment**

**July 2023 Presentation**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **Lau Wei Lian** | M2273019 |

**Tutorial Group: ­­­­­­­­­­­­ T03**

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

**Submission Date: 14/09/2023**

Table of Contents

[Question 1(a) 3](#_Toc145627317)

[Question 1(b) 4](#_Toc145627318)

[Question 1(c) 5](#_Toc145627319)

[Question 2 7](#_Toc145627320)

[References 8](#_Toc145627321)

# Question 1(a)

Plagiarism is an issue that has severe consequences to the overall academic integrity. Coding is a complex subject for any beginner; thus, with internet access, finding readily available code online and passing it off as one's work is effortless.

Reasons for plagiarism are as follows:

**Poor understanding:** Students may need help grasping the coding concept and, for whatever reason, go the easy way out of copying other students' code.

**Ease of Internet:**Codes are readily available online, and with the Internet available 24/7, it's effortless to get someone's code anytime, anywhere.

How to overcome coding plagiarism:

**Additional effort on Education:** The students and the school must go the extra mile to ensure all students can grasp the basic coding concept before going deeper into the topic.

**Uncommon Assignments:**The requirements to have real-life scenario questions encourage students to work on a solution rather than look for answers online where codings are readily available.

In conclusion, it is the need for additional education that will help reduce the possibility of plagiarism.

Word count – 170 words

# Question 1(b)

people = 20

cates = 30

dogs = 15

if people < cats:

print "Too many cats! The world is doomed!"

if people < cats:

print("Not many cats! The world is saved!")

if people < dogs:

print("The world is drooled on!")

if people > dogs

print("The world is dry!")

dogs += 5

if people >= dogs:

print("People are greater than or equal to dogs.")

if people <= dogs

print("People are less than or equal to dogs.)

if people = dogs:

print("People are dogs.")

(A.Shaw, 2017)

The above code is part of a study drill on introducing the if- statement in Python.

The first half of the codes have three pre-set numbered variables: people, cats and dogs.

Next, "if" statements in the coding portion were programmed to generate pre-set strings if one variable is more than another.

Hence, pre-set string number one would be reflected if condition number one is fulfilled.

Should condition number one not be completed, the code would run the second condition, and again, if not fulfilled, it would run the third and final conditions.

The second half of the codes have only one variable, while the "if" statements are programmed to run three different conditions.

Word count – 113 words

# Question 1(c)

people = []

cats = []

dogs = []

while True:

try:

people\_input = int(input(“Enter the number of people or ‘stop’ to exit:”)

if people\_input.lower() == “stop” :

break

people\_input = int(people\_input)

cats\_input = int(input(“Enter the number of cats”)

dogs\_input = int(input(“Enter the number of dogs”)

except ValueError:

print(“Invalid input. Please enter a number.”)

continue

people.append(people\_input)

cats.append(cats\_input)

dogs.append(dogs\_input)

if people < cats:

print ("Too many cats! The world is doomed!")

if people < cats:

print("Not many cats! The world is saved!")

if people < dogs:

print("The world is drooled on!")

if people > dogs

print("The world is dry!")

dogs\_input += 5

if people >= dogs:

print("People are greater than or equal to dogs.")

if people <= dogs

print("People are less than or equal to dogs.)

if people = dogs:

print("People are dogs.")

The program has been rewritten, upgrading from a simple "if" statement coding to include looping, user input, integer input and list.

The first thing was to create a list for the three variables.

The user's input is now stored in the list, allowing the programmer to track the iterations if required. These iterations would create the base for data analysis if required.

Adding the looping command allows the user to continue the program indefinitely or quit by inputting the keyword, stop. Controls are included to check whether the user input is an integer or variable, as the first user input requires an integer to move to the second input. At the same time, if it's not an integer, it needs to be specific input to break the loop.

The original code is programmed to complete only one run; we have expanded the program to run multiple times with different permutations based on the user inputs.

This revised code gives the user more flexibility to run the program with significantly more permutations.

It's possible because we have added options to have a list to store user input, which would allow the user to stretch the codes to provide the appropriate message based on different user inputs and conditions met.

The original code consists of pre-fixed variables, while the revised code now runs on user-defined variables.

Word count – 224 words

# Question 2

Appendix 1 has been improved as per the new code below:

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

entered\_input ={}

updated\_items =[]

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

while True:

item = (input("Hello! What do you want to buy? "))

if item not in products:

print(f'{item} is an invalid Product! Please try again.')

continue

price\_of\_item = input(f"How much is the {item} (in SGD)? ")

try :

price\_of\_item = int(price\_of\_item)

except ValueError :

print("Invalid input. Please enter a valid price.")

continue

entered\_input[item] = price\_of\_item

updated\_items.append((item, price\_of\_item))

f\_input = input("Would you like to continue ? (yes/no)")

if f\_input.lower() == "no":

break

print(f'This is your updated shopping list: {updated\_items}')

In the above-revised coding, I have included a 'while True" loop for a continuous flow of commands and to improve the overall reliability.

The revised code allows the user to continue the flow of commands until the user decides it's time to finish shopping. The loop commands improve the previous set, where the code stops once an error occurs.

Included is the definition of integer in the input of value under the command of how much the items are. Should the user input something non-integer, the code would prompt the user to amend accordingly.

This change also improves the code's quality and readability from both the creator's and user's perspectives.

Word count – 121 words

# References

A.Shaw, Z. (2017). *Learn Python 3 The Hard Way.* United States of America: Pearson.

Arora, S. K., & Johns, R. (2023, Sep 11). *30 Cool, Easy & Fun Python Projects with Source Code [2023]*. Retrieved from hackr.io/: https://hackr.io/blog/python-projects#:~:text=Dice%20Roll%20Generator&text=As%20one%20of%20the%20Python,you've%20rolled%20the%20dice.