****

**ANL252**

**Python for Data Analytics**

Tutor-Marked Assignment 01

July 2023

Submitted by:

Ang Xiu Zhi

J1771750

Content Page

|  |  |
| --- | --- |
| Content | Page No. |
| Question 1a | 2 |
| Question 1b | 2 |
| Question 1c  Question 2 | 4  6 |
| References | 9 |

Question 1a

Plagiarism in coding refers to copying of codes without valid authorization or credits to the original code (Copyleaks, 2021).

* The primary reason that causes coding plagiarism to happen is due to referencing of source codes from public repositories. Very often when referencing from public repositories, the user fails to give credit to the source since this is not being enforced outside of education institutes.
* Ambiguity of collaboration also give rise to plagiarism. Collaboration which is common in software development are likely unacceptable in academic class (Lee, 2020) since the purpose is to assess individually.

Some ways to avoid coding plagiarism as follow:

* Building the code on your own will help to avoid plagiarism almost entirely since the entire code is original and made by yourself. By coming up with unique codes eliminates the chances of a plagiarism strike.
* Be clear on what external codes can be used and give proper referencing. Check on what are the external sources that are allowed to be used for the coding work and ensure that proper referencing has been given either in the main file or within the reference section.
* Using of plagiarism checker to check for coding plagiarism.

Question 1b

The python code below ascertains whether the name is registered in the name list set before. By using list instead of tuples allows the object within the list to be modified. If the user enters the correct name as registered within the list, he will proceed to the next section. If a wrong name is being entered, the code will prompt the user to re-enter the correct name or type quit to exit the code.

In the next section, once the name has been ascertained to be registered in the list, he will then be prompted to enter in the age, gender, height and weight. Data validation has been applied to the gender field so if any other value is entered the code will not proceed. Once data is received, the BMI of the user is then calculated, and the code will generate a text file in the same directory. The data entered above will then be saved into the text file and the code will inform the user on his BMI number and the name of the text file. The user can then proceed to enter information on the other registered user or type “quit” to exit the code.

# Name list of participants

name\_list = ["aiden", "yvonne", "jieting", "xinyi", "syafiq"]

# Start the user interaction loop

while True:

# Request for name of participant

user\_name = input("Enter participant's name (or press Enter to exit): ").lower()

# Exit loop for program

if user\_name.lower() == "":

break

# Request for the following variable

if user\_name in name\_list:

user\_age = input("Enter your age: ")

while True:

user\_gender = str(input("Enter participant’s gender (M/F): "))

if user\_gender in ("M", "F"): # Ensures that the gender input is correct

break

else:

print("Please enter either M or F.")

user\_height = float(input("Enter participant’s height (in cm): "))

user\_weight = float(input("Enter participant’s weight (in kg): "))

# Calculate the BMI of participant

user\_BMI = round(((user\_weight) / ((user\_height/100)\*\*2)),1)

# Ensure that participant's name is capitalized

cap\_name = user\_name.capitalize()

# Create a text file

file\_name= f"{cap\_name}.txt"

with open(file\_name, "w") as file:

file.write(f"Name: {cap\_name}\n")

file.write(f"Age: {user\_age}\n")

file.write(f"Gender: {user\_gender}\n")

file.write(f"Height (cm): {user\_height}\n")

file.write(f"Weight (kg): {user\_weight}\n")

file.write(f"BMI: {user\_BMI}\n")

print(f"Participant’s BMI is {user\_BMI}. Additional information has been updated and saved to {file\_name}")

else:

# Request for new name

print("Participant’s name not found in the list. Please enter a correct name.")

print("participant information save, please find the respective text files in the same directory")

Question 1c

In order to avoid coding plagiarism, there are three ways to rewrite the python code:

1. Use more precise comments.

All programming languages provide the users with a way to insert comments into their code. Commenting code is a good practice every programmer should follow (Bailey, 2017). However, comments since comments are specific to each individual and the usage of the code, coding incorporated from outside sources are usually not original. Therefore, comments within the code should reflect what the purpose of the code is. More precise and individualised comments also allow the user to understand what the particular piece of coding is used for so easier edits can be done.

1. Unique variable names

Changing of the variable names does not make the code original (Bailey, 2017). However, it is best to ensure that the variable and other names used within the code be as original as possible. By having original names which are specific to the piece of code makes coding and modifying the code much easier since it helps in describing what the variable is being used for. Having original names also helps in highlighting the originality of the piece of code since it is more personalised.

1. Using more function parameters

Function parameters helps in making the code more precise and allows for better data validation. By adding in more function parameters such as the try and except function helps to prevent error data from being entered. Function also helps in making the code more personalised since it helps the code cater more closely to the need of the coder in comparison to grabbing code from an outside source.

# Name list in lower case

nominal\_roll = ["aiden", "yvonne", "jieting", "xinyi", "syafiq"]

# Start the user interaction loop

while True:

# Request for name of participant

participant\_name = input("Enter participant's name (or press Enter to exit): ").lower()

# Exit loop for program

if participant\_name.lower() == "":

break

# If the name is in the list, ask for the following variable

if participant\_name in nominal\_roll:

while True:

try:

participant\_age = float(input("Enter participant’s age: "))

break

except ValueError:

print("Please enter a valid number.")

while True:

participant\_gender = str(input("Enter participant’s gender (M/F): "))

if participant\_gender in ("M", "F"): # Ensures that the gender input is correct

break

else:

print("Please enter either M or F.")

while True:

try:

participant\_height = float(input("Enter participant’s height (in cm): "))

break

except ValueError:

print("Please enter a valid number.")

while True:

try:

participant\_weight = float(input("Enter participant’s weight (in kg): "))

break

except ValueError:

print("Please enter a valid number.")

# Calculate the BMI of participant & round to 1 decimal place

participant\_BMI = round(((participant\_weight) / ((participant\_height/100)\*\*2)),1)

# Create a text file with the participant's name as the filename

file\_name= f"{participant\_name.capitalize()}.txt"

with open(file\_name, "w") as working\_file:

working\_file.write(f"Participant name: {participant\_name.capitalize()}\n")

working\_file.write(f"Age: {participant\_age}\n")

working\_file.write(f"Gender: {participant\_gender}\n")

working\_file.write(f"Height (cm): {participant\_height}\n")

working\_file.write(f"Weight (kg): {participant\_weight}\n")

working\_file.write(f"BMI: {participant\_BMI}\n")

print(f"Participant’s BMI is {participant\_BMI}. Additional information has been updated and saved to {file\_name}")

else:

# Request for a correct name if a wrong one is entered

print("Participant's name not found in the list. Please enter a correct name.")

print("User information save, please find the respective text files in the same directory")

Output generated:

Enter participant's name (or press Enter to exit): aiden

Enter your age: 29

Enter your gender (M/F): M

Enter your height (in cm): 174

Enter your weight (in kg): 83

Your BMI is 27.4. Additional information has been updated and saved to Aiden.txt

Enter participant's name (or press Enter to exit):

User information save, please find the respective text files in the same directory

Question 2

The three points that can help enhance the code is as follows:

1. Error handling

By implementing error handling into the code helps in ensuring the reliability and quality of the output since it helps in data validation, such as for the price and query. If the value being entered is inappropriate, the code will request a new entry to be entered again to ensure that it is correct before proceeding, ensuring the output is more reliable.

1. Descriptive variable names

In this code, previously the variable name of ‘item’ and ‘products” is used. This variable name is more general in comparison to the new variable name ‘intended\_purchase’ and ‘product\_list’. By changing this variable name helps enhance the readability since it is clearer what the variable represents.

1. Proper formatting

Having a proper format also helps in enhancing the readability of the code. It is good practice to add comments for the codes. This enhances the readability of the code by helping to explaining what each individual code and sections within the code does. Adding comment helps make editing the code since it helps to make singling out certain functions much easier. Use of space between lines and around operators also makes the code more readable since the code is spread out.

# Defines the products contained within the list

product\_list = ['laptop', 'mouse', 'webcam', 'keyboard', 'speaker']

# Initialize the query variable

query = 'yes'

# Initialize the list for the shopping list

shopping\_list = []

# Display the list of products available

print(f"We have these products currently available: {product\_list}.")

# Start a loop to interact with the user

while query == 'yes':

intended\_purchase = str(input("Hello! What do you want to buy? "))

# Check if item intended for purchase is in the list of products

if intended\_purchase not in product\_list:

print(f"Product currently not available! Please enter another product.")

break

# Requesting for the price of the intended purchase

while True:

# Checking the price for numerical error

try:

item\_price = float(input("How much is it (in SGD)? "))

break

except ValueError:

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

entered\_input = [intended\_purchase, item\_price]

shopping\_list.append(entered\_input)

# Checking to see if customer wishes to continue

while True:

query = str(input("Would you like to continue? (yes/no)"))

if query in ('yes', 'no'):

break

else:

print("Please enter yes or no")

# Display the updated shopping list

print(f"This is the updated shopping list: {shopping\_list}")

References

Bailey J. (Nov 8, 2017). Plagiarism and Programming: How to Code Without Plagiarizing. Retrieved from: <https://turnitin.com/blog/plagiarism-and-programming-how-to-code-without-plagiarizing-2>

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

Lee C. (Jul 28, 2020). What is Programming Plagiarism? Why Is It on the Rise?. Retrieved from: <https://www.turnitin.com/blog/what-is-programming-plagiarism-why-is-it-on-the-rise>