

**ANL252 Python for Data Analytics**

# **Tutor-Marked Assignment**

**July 2023**

Submitted by: Nur Ruzainatul Husnaa Binte Abdul Jalil (2310679)

Tutorial Group: ­­­­­­­­­­­­ T TO3

Instructor: Dr. Munish Kumar

Submission Date: 15/09/2023

**QUESTION 1 (a)**

Plagiarism in coding happens when using another person's codes and making them yours by simply copying and pasting them. There are many reasons for plagiarism in coding and some reasons why it happens.

* Collaboration is the nature of programming, and almost all computer programs contain many ideas borrowed from somewhere else.
* The codes are easily access and free for anyone to use online.
* Time pressure, especially for a part-time student like us with a full-time work commitment, while some of us have families to care for.
* Unfamiliar with plagiarism in coding as it is different from the regular essay-based assignments.

How to avoid plagiarism issues by

* Give credit to the original writer or source online by acknowledging their codes and referencing.
* Use code repositories and online sources responsibly, and do not write their codes as your own.
* Plan and start early on your assignments to allow yourself to revise and have plenty of time to complete the tasks.
* Understand the concepts by ensuring you fully understand the coding concepts you are asked to apply.

**QUESTION 1 (b)**

finallist = []

i=i

proceed = True

print("\nPlease type in the following data (Press ENTER to exit):")

while proceed == True:

std\_name = input("Name of the student (i): ")

if std\_name == "";

proceed = False

break

valid\_input = False

while valid input == False:

std\_score = input(f" (std\_name}\'s score in the exam: ")

if std\_score== "";

proceed = False

break

try:

std\_score = float(std\_score)

except valueError:

print("Your input is not numeric. Please try again")

else:

valid\_input = True

if proceed == True:

templist = [std\_name, std\_score]

finallist = finallist + [templist]

i=i+1

if i>1:

print("\nThe following data have been stored:\n{finallist}\n")

Please type in the following data (Press ENTER to exit):

Name of the student (1): Peter

Peter's score in the exam: test

Your input is not numeric. Please try again

Peter's score in the exam: 72

Name of the student (1): Mary

Mary's score in the exam: 86

Name of the student {1}:

The following data have been stored: [['Peter', 72.0], ['Mary', 86.0]]

Source: Study Guide ANL252 Phyton for data analytics

The codes get the user to input the student's name and score of the student and store the information input in a score list. First, the code creates an empty list named “finallist” to keep the information input by the students. Then while the loop is used so that the students input their name and score, the loop will break if the user enters an empty string for their name. If the students enter a valid name and score, the code will create a temporary list containing their name and score and add it to the brief list called “finallist.” Finally, the code prints the “finallist” list.

**QUESTION 1 (c)**

# Create an empty list called “finalist” to store the input.

* Set a variable “i” to 1
* Set the variable proceed to “True”

# Print a message to type in student information

* While loop will continue to iterate as long as the proceed variable is “True”
* Input the student’s name and store it in the std\_name variable. Print “Name of the student (1)”
* If they input an empty string, proceed variable will be “False” and break the while loop
* Set valid\_input variable as “False”.
* While loop will continue as long as valid\_input variable is “False”.
* Input the student’s score and keep it in std\_score variable
* If empty string is input for student’s score, proceed variable will set to “False” and break the while loop.
* Convert the student’s score to a floating number
* If failed print “Your input is not numeric. Please try again”
* If its successful, set valid\_input to “True”
* Break the while loop.

# Add the student's information to a list.

* If the proceed variable is “True”, create a temporary list to “finalist” list.
* If i variable is greater than 1, print the “finalist” list and print “The following data have been stored”.

There are many reasons why it is essential to rewrite rules with succinct bullet points besides ​​avoiding plagiarism. The advantage of bullet points over simplicity is that they provide a compelling precedent reading. Bullet points make it easy for people to find and understand code quickly, useful for debugging and code review. Additionally, bullet points can increase code brevity by eliminating unnecessary code, simplifying it, and making it easier for the reader to understand the code. Besides, bullet points can also help improve code readability because they can create related and unrelated blocks of code. Combining code can help clarify the regulation flow and each section's purpose. Finally, breaking code into smaller pieces makes it easier to understand and debug. Bullet points can also help reduce complexity by simplifying code. Last but not least, bullet points are a process that can help your code remain maintainable by making it easy to add, remove, and modify your code without damaging other parts of the code.

**Question 2**

def get\_product\_input():

"""Get the user's input for the product they want to buy.

Returns:

A str representing the product name.

"""

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

for product in products:

print(f"Would you like to buy a {product}? (yes/no)")

user\_input = input()

if user\_input == "yes":

return product

print("Invalid product selection.")

return None

def add\_product\_to\_updated\_list(updated\_list, product, price):

"""Add a product to the updated shopping list.

Args:

updated\_list: A list of lists, where each inner list represents a product

in the shopping list.

product: A str representing the product name.

price: A float representing the product price.

"""

updated\_list.append([product, price])

def print\_updated\_list(updated\_list):

"""Print the updated shopping list.

Args:

updated\_list: A list of lists, where each inner list represents a product

in the shopping list.

"""

print("Updated shopping list:")

for product, price in updated\_list:

print(f"{product}: {price}")

if \_\_name\_\_ == "\_\_main\_\_":

updated\_list = []

while True:

product = get\_product\_input()

if product is None:

break

price = float(input("How much is it? "))

add\_product\_to\_updated\_list(updated\_list, product, price)

print\_updated\_list(updated\_list)

There are multiple ways to improve the codes, and here are some of them. Break them into small chunks by breaking them into input groups, checking the product in the list group, adding the product to the updated list group, and printing the updated list. Next, add type notations. Adding type notations can help make the code more readable. For example, item and price\_of\_item variables could be specified as str and float. Use a loop to get the user’s input. The current code uses a while loop to get the user’s information on the item he wants to buy. However, this loop will continue to execute again even if the user enters an item not in the list. It would be better to use a for loop to prompt the user to select an item from the list of things to iterate. The current code does not handle invalid user input. For example, if the user enters an item not on the list, the code will print a message and then break the loop. It is best to handle incorrect user input nicely and prompt the user to try again by adding invalid input.

**Reference:**

*Avoiding plagiarism | Oregon Tech*. (n.d.). <https://www.oit.edu/library/help/avoiding-plagiarism>

Lee, C. (2023, September 15). 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#:~:text=Source%20code%20plagiarism%E2%80%94otherwise%20known,as%20your%20own%20without%20attribution>.

*The importance of clear and concise bullet points on a landing page*. (n.d.). <https://www.markettailor.io/blog/importance-of-clear-and-concise-bullet-points-on-landing-page>

Wu, K. Y., & Zhu, S. (2023). *ANL252 Python for data analytics.* Singapore University of Social Science.