

**ANL252**

**Tutor-Marked Assignments**

**H2311509**

**Shafina Binte Hazman**

**15 September 2023**

**Question 1**

**(a) Reasons for Plagiarism in coding:**

* **Lack of understanding:**

Students may copy code when they don't understand the concepts.

* **Time pressure:**

Tight deadlines can lead to copying code from online sources.

* **Academic pressure:**

Fear of failure or competition can drive students to cheat.

(Network World, 2012)

**Avoiding Plagiarism:**

* **Cite Sources:**

Properly credit and reference any code you borrow.

* **Time Management:**

Plan assignments well in advance to avoid last-minute copying.

* **Use Plagiarism Detection Tools:**

Employ tools like Turnitin to check for code similarities.

* **Embrace Independent Coding for Learning:**

Achieve effective learning by overcoming challenges and errors.

Through independent problem-solving, students can discover solutions and refine skills.

* **Seek Help:**

If students are struggling, they can ask for assistance from professors or tutors.

(Copyleaks, n.d.)

**(b) Python Code Example:**

exchange\_rates = {

"USD": 1.0,

"EUR": 0.85,

"GBP": 0.73,

}

def convert\_currency(amount, from\_currency, to\_currency):

if from\_currency in exchange\_rates and to\_currency in exchange\_rates:

conversion\_rate = exchange\_rates[to\_currency] / exchange\_rates[from\_currency]

converted\_amount = amount \* conversion\_rate

return converted\_amount

else:

return "Currency not found in exchange rates."

amount = float(input("Enter the amount: "))

from\_currency = input("From Currency (3-letter code, e.g., USD): ").upper()

to\_currency = input("To Currency (3-letter code, e.g., EUR): ").upper()

result = convert\_currency(amount, from\_currency, to\_currency)

if isinstance(result, str):

print(result)

else:

print(f"{amount} {from\_currency} is equal to {result} {to\_currency}")

With the Python code provided, an input amount can be converted from one currency to another without using external libraries. It prompts the user to enter the amount, the source currency (e.g., USD), and the target currency (e.g., EUR).

This code defines exchange rates for various currencies manually and converts them based on a predefined dictionary. In order to calculate the converted amount, the target currency's rate is divided by the source currency's rate, then multiplied by this rate. After the conversion has been completed, the program displays the original and converted amounts. (OpenAI, 2023).

**(c) Rewritten Python code with an explanation:**

exchange\_rates = {

"USD": 1.0,

"EUR": 0.85,

"GBP": 0.73,

}

def convert\_currency(amount, from\_currency, to\_currency):

if from\_currency in exchange\_rates and to\_currency in exchange\_rates:

conversion\_rate = exchange\_rates[to\_currency] / exchange\_rates[from\_currency]

converted\_amount = amount \* conversion\_rate

return converted\_amount

else:

return "Currency not found in exchange rates."

amount = 100 # Replace with the desired amount

from\_currency = "USD" # Replace with the desired source currency

to\_currency = "EUR" # Replace with the desired target currency

result = convert\_currency(amount, from\_currency, to\_currency)

if isinstance(result, str):

print(result)

else:

print(f"{amount} {from\_currency} is equal to {result} {to\_currency}")

100 USD is equal to 85.0 EUR

**Rationale for rewrite:**

* **Static Conversion:**

It is designed for static currency conversion. Users need to edit the code itself to change the conversion parameters.

* **Batch Processing:**

Converting a fixed amount from one currency to another without the help of the user is suitable for this method. It's ideal for scenarios where you have a list of predetermined conversions to perform. (Barone, 2021)

* **Automated Output:**

It directly prints the result without asking for user input or handling user errors. It's best when you want to automate the conversion process and just need the final results to be displayed without user interaction. (Sundnes, 2020)

**Question 2**

**An improved version of Appendix 1:**

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

query = 'yes'

updated\_items = []

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

while query == 'yes':

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

if item not in products:

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

break

price\_of\_item = input("How much is it (in SGD)? ")

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

updated\_items.append(entered\_input)

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

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

**Ways to enhance its reliability, readability, maintainability, and overall quality:**

**1. Fix Syntax Errors:**

* There is a syntax error in the products list. The string 'keyboard' is missing an opening single quote.
* In the query variable assignment, it should be query = 'yes' (with single quotes) instead of query = 'yes' (with double quotes).
* The input function is misspelt as imput. Fix it to input.

**2. Use Meaningful Variable Names:**

* Use more descriptive variable names to improve readability. For example, replace item with product\_name and entered\_input with product\_info.

**3. Handle Input Case Sensitivity:**

* Make the product input case-insensitive by converting the user's input to lowercase before checking it against the products list. This ensures that "Laptop" and "laptop" are treated as the same product.

**References:**

Barone, A. (2021, August 04). Batch Processing. Investopedia. Retrieved from <https://www.investopedia.com/terms/b/batch-processing.asp>

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

Network World. (2012). Why computer science students cheat. Retrieved from <https://www.networkworld.com/article/2207189/why-computer-science-students-cheat.html>

OpenAI. (2023). GPT-3.5-based ChatGPT (Version 3.5) [Large language model]. <https://chat.openai/com/chat>

Sundnes, J. (2020). Introduction to Scientific Programming with Python (Simula SpringerBriefs on Computing 6). Springer.