

## Assignment Title Page

|  |  |
| --- | --- |
| **STUDENT NAME:** | Hilmi Bin Ishak |
| **P.I. NO.:** | N2181344 |
| **TITLE:** | TMA01 |
| **COURSE CODE:** | ANL252 |
| **TG:** | T03 |
| **SUBMISSION DATE:** | 13/9/2023 |

**Plagiarism and Collusion**

**Plagiarism** is the act of using or passing as one’s own, the ideas or writings of another without acknowledging or crediting the source from which the ideas are taken from.

**Collusion** is the act of submitting any academic work (including assignment, project or report) that was completed by another person and pass these work off as one’s own.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Declaration** | | | | |
|  | | | | |
| I declare that this assignment is my own work, unless otherwise acknowledge or credited by appropriate referencing. I have read and abide by the SUSS Honour Code and I am aware of the penalties associated with plagiarism and collusion listed in the Student Handbook. | | | | |
|  | | | | |
| Initial: | Hilmi Ishak |  | Date: | 13/9/2023 |
|  | | | | |

**Question 1**

a)

Plagiarism in coding happens when one takes a code that was done by others and passing it as their own work. (Lee, 2021) With codes being available online through various websites, students can simply look up such codes and copy them into their work. When they do so, some would simply take the entire code and submit as is without making any form of change to the code nor do they even acknowledge the source where or who the code is from. (Cosma & Joy, 2008)

In order to prevent such plagiarism, there are certain steps we can take such as:

* Remembering to always cite your sources and when using codes done by others, give them due credit by including the author’s name as well as the website that you found the code
* Receiving permission from the author themselves
* Use your own style when it comes to coding. This allows you to know which codes are yours and which is not. The ones that are not yours should be cited or credited. (Mercado, 2022)

Word count: 177 words including in-text citations

b)

weight = float(input("Enter your weight in pounds: "))

height = float(input("Enter your height in inches: "))

age = int(input("Enter your age: "))

sex = input("Enter your sex (male or female): ")

activity\_level = input("Enter your activity level (sedentary, lightly active, moderately active, very active): ")

def calculate\_bmr(weight, height, age, sex):

if sex == "male":

bmr = 66 + (6.3 \* weight) + (12.9 \* height) - (6.8 \* age)

else:

bmr = 655 + (4.3 \* weight) + (4.7 \* height) - (4.7 \* age)

return bmr

def calculate\_daily\_calories(bmr, activity\_level):

if activity\_level == "sedentary":

calories = bmr \* 1.2

elif activity\_level == "lightly active":

calories = bmr \* 1.375

elif activity\_level == "moderately active":

calories = bmr \* 1.55

else:

calories = bmr \* 1.725

return calories

bmr = calculate\_bmr(weight, height, age, sex)

calories = calculate\_daily\_calories(bmr, activity\_level)

print("Your daily calorie needs are: ", calories)

Code sourced from: <https://github.com/KoderKumar/Calorie-Intake-Calculator/blob/main/main.py>

Author: KoderKumar

Date retrieved: 12 September 2023

The code above is a calorie intake calculator. It firstly asks for the user’s details such as their age, weight, height, gender and their level of activity. From there, the code will calculate the basal metabolic rate (BMR) according to gender via an if-else loop before going through an if-elif-else loop which loops according to their activity level. Lastly, the user will then be shown what their daily caloric intake is. Mathematic calculations are required in order to calculate the BMR. For this instance, the BMR is calculated using the Harris-Benedict equation. (Zając & Mucha, n.d.)

This calorie intake calculator allows users to know how much calories that they are needed to consume each day according to their activity level to which from the results, they can then follow said calorie intake in order to maintain their body weight. Such results can also give users a rough idea on how they can achieve body goals that they may have as either lesser could reduce their weight whereas eating more could make them gain weight over time. (MedicalNewsToday, n.d.)

Word count: 178 words including in-text citations

c)

Code:

# Gathering information

print("Welcome to the calorie and BMI calculator!\n\nIf given choices, do type it word for word (Case sensitive)\n")

weight = int(input("Weight in KG: "))

height = int(input("Height in cm: "))

age = int(input("Age: "))

gender = str(input("Male or Female: "))

activity = str(input("Activity level (Sedentary, Lightly active, Moderately active, Active or Very active): "))

print(f"\nYou have inserted the following: {weight}, {height}, {age}, {gender}, {activity}")

# Confirmation of input

confirm = str(input("\nContinue to calculate? (Yes or No): "))

if confirm == "No":

print("Thank you for using this calculator!")

else:

print("\nHere are your results:")

# BMR and Calories Calculation

def calculatedBMR(weight, height, age, gender):

if gender == "Male":

BMR = (66.5 + (13.75 \* weight) + (5.003 \* height) - (6.75 \* age))

else:

BMR = (655.1 + (9.563 \* weight) + (1.850 \* height) - (4.676 \* age))

return BMR

BMR = calculatedBMR(weight, height, age, gender)

def calculated\_calories(BMR, activity):

if activity == "Sedendary":

calories = BMR \* 1.2

elif activity == "Lightly active":

calories = BMR \* 1.375

elif activity == "Moderately active":

calories = BMR \* 1.55

elif activity == "Active":

calories = BMR \* 1.725

else:

calories = BMR \* 1.9

return calories

calories = calculated\_calories(BMR, activity)

# BMI Calculation, Grouping and Results

BMI = weight/((height/100) \* (height/100))

if BMI < 18.5:

print("\nYou are underweight category")

elif BMI > 18.5 and BMI < 22.9:

print("\nYou are in the normal weight category")

elif BMI > 23 and BMI < 29.9:

print("\nYou are in the overweight category")

else:

print("\nYou are in the obese category")

print(f"Your calorie intake is: {round(calories)}kcal")

rec\_carb = round((calories \* 0.50)/4)

rec\_pro = round((calories \* 0.30)/4)

rec\_fats = round((calories \* 0.20)/9)

print(f"\nHere is your recommended calorie breakdown:\

\n{rec\_carb}g of Carbohydrates\

\n{rec\_pro}g of Proteins\

\n{rec\_fats}g of Fats")

Output:

Welcome to the calorie and BMI calculator!

If given choices, do type it word for word (Case sensitive)

Weight in KG: 60

Height in cm: 165

Age: 26

Male or Female: Male

Activity level (Sedentary, Lightly active, Moderately active, Active or Very active): Sedentary

You have inserted the following: 60, 165, 26, Male, Sedentary

Continue to calculate? (Yes or No): Yes

Here are your results:

You are in the normal weight category

Your calorie intake is: 2929kcal

Here is your recommended calorie breakdown:

366g of Carbohydrates

220g of Proteins

65g of Fats

The code provided in the previous part was rewritten and new features have been added to it, such as a BMI calculator as well as a calorie breakdown. The new features allows the user to get a more in-depth result and thus would allow them to achieve their body goals better, such as knowing their weight category, to which they can either lose or gain weight if needed. Should they want to maintain to maintain their weight, the calorie breakdown allows them to know what they should intake daily and how much of it as well.

Thus, the rationale of rewriting the code would be as such:

* Rewriting of the codes would allow us to personalize the codes, as instead of simply copying and pasting a whole chunk of code from another, we can rewrite it according to our style, such as formatting how the codes look like or even different functions which provides the same result. Other forms of personalizing could mean reformatting the metrics used to one that is more familiar to us, such as from inches to centimetres
* On top of customizing it, we can also add on to the code such as add new features would make the code even more meaningful for users
* Ultimately, rewriting the code allows the code to be more tailored to our needs and also thus allow us to avoid plagiarism by copying the author of the code word by word.

Word Count: 243 words

**Question 2**

The first change that could be made is instead of using a list for the products, we can use a dictionary instead. Using a dictionary allows the admin to input the prices of the items and would increase the reliability of the code as the admin can update the items according to what stock they have in store. The second change would be to allow users to see the stocks that are available to purchase. This allows customers to type the price according to what is shown instead of a random price of an item to which the store might not have that item with that price in stock. The last change would be the to have a formatted integer input when the user is asked to input the price of the item that they would like to purchase. Additionally, breaks are also added into the code to enhance readability of the output.

Word count: 153

Edited code:

products = {'laptop':[1000, 2000, 3000], 'mouse':[20, 50, 80], 'webcam':[80], 'keyboard':[100], 'speaker':[150]}

query = 'yes'

updated\_items = []

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

while query == 'yes':

item = str(input("\nHello! What do you want to buy? "))

if item not in products:

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

break

item\_value = products.get(item)

print(f"\nHere are your {item} options: {item\_value}")

price\_of\_item = int(input("\nHow much is the item you would like to purchase (in SGD)? "))

if price\_of\_item not in item\_value:

print(f'Item is not found')

break

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

updated\_items.append(entered\_input)

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

Output:

We have a list of products here: dict\_keys(['laptop', 'mouse', 'webcam', 'keyboard', 'speaker']).

Hello! What do you want to buy? laptop

Here are your laptop options: [1000, 2000, 3000]

How much is the item you would like to purchase (in SGD)? 1000

Would you like to continue? (yes/no) no

This is our updated shopping list: [['laptop', 1000]]

# Bibliography

Lee, C. (2021, July 14). *What is source code plagiarism and what does it have to do with academic integrity?* Retrieved from Turnitin: https://www.turnitin.com/blog/what-is-source-code-plagiarism-and-what-does-it-have-to-do-with-academic-integrity

Cosma, G., & Joy, M. (2008). *Towards a Definition of Source-Code Plagiarism.* IEEE TRANSACTIONS ON EDUCATION.

Mercado, K. (2022, July 8). *Does Plagiarism Issue Apply To Programming?* Retrieved from Kode Java: https://kodejava.org/does-plagiarism-issue-apply-to-programming/

Zając, A., & Mucha, M. (n.d.). *Harris-Benedict Calculator (Basal Metabolic Rate)*. Retrieved from Omni Calculator: https://www.omnicalculator.com/health/bmr-harris-benedict-equation

MedicalNewsToday. (n.d.). *How many calories should I eat a day?* Retrieved from MedicalNewsToday: https://www.medicalnewstoday.com/articles/245588