Brandon Tonge
ID – 201358937

COMP101 – Assignment 03

Python Code –

```
# 201358937 Tonge_Brandon-CA03.py
# October 2018
# This program uses a menu system to navigate the user to a nutrition adviser.
# This will accept the users inputs and then outputs the colour rating as per
# the Food Standards Agency.

# Main Function
def main():
    print("---Main Menu---")
    print("A - Selection")
    print("B - Iteration")
    print("C - Games")
    print("E - Extend")
    print("X - Exit Program")
    print("")
    choice = str.upper(input("Please select an option from the menu: "))

    # TEST
    # print(choice)

    # Function Selection
    if(choice == "A"):
        selection()

    elif(choice == "B" or choice == "C"):
        development()

    elif(choice == "E"):
        extended()

    elif(choice == "X"):
        exit()

    else:
        print("\nPlease enter a valid choice!\n")
        main()


#Selection Function
def selection():
    # User inputs
    print("\nPlease enter the values per 100g:")
    fat = float(input("Fat: "))
    saturates = float(input("Saturates: "))
    sugar = float(input("Sugar: "))
```

Brandon Tonge
ID – 201358937

```
    salt = float(input("Salt: "))
    portion = float(input("Portion size in grams: "))

    # Fat Test
    if(fat <= 3):
        fat_class = "Green"

    elif(fat>3 and fat<=20):
        fat_class = "Amber"

    else:
        fat_class = "Red"

    # Saturates Test
    if(saturates <= 1.5):
        saturates_class = "Green"

    elif(saturates>1.5 and saturates<=5):
        saturates_class = "Amber"

    else:
        saturates_class = "Red"

    # Sugar Test
    if(sugar <= 5):
        sugar_class = "Green"

    elif(sugar>5 and sugar<=12.5):
        sugar_class = "Amber"

    else:
        sugar_class = "Red"

    # Salt Test
    if(salt <= 0.3):
        salt_class = "Green"

    elif(salt>0.3 and salt<=1.5):
        salt_class = "Amber"

    else:
        salt_class = "Red"

    # Fat Portion
    if(fat * (portion/100)>21):
        fat_class = "Red (Portion)"

    # Saturates Portion
    if(saturates * (portion/100)>6):
        saturates_class = "Red (Portion)"
```

Brandon Tonge
ID – 201358937

```
    # Sugar Portion
    if(sugar *(portion/100)>15):
        sugar_class = "Red (Portion)"

    # Salt Portion
    if(salt *(portion/100)>2.4):
        salt_class = "Red (Portion)"

    # Outputs
    print(f"\nPortion size = {portion}g")
    print(f"{fat_class}: Fat {fat}g per 100g")
    print(f"{saturates_class}: Fat {saturates}g per 100g")
    print(f"{sugar_class}: Fat {sugar}g per 100g")
    print(f"{salt_class}: Fat {salt}g per 100g")
    print()



    '"TEST
    print(fat_class)
    print(saturates_class)
    print(sugar_class)
    print(salt_class)
    print(fat,saturates,sugar, salt, portion)"'

    main()

# Development Function
def development():
    print("\nFunction under development\n")
    main()

# Extended Function
def extended():
    print("\nExtended Requirements\n")
    main()

main()
```

Testing Table –

Inputs	Expected Output	Actual Output	Comments
Fat – 3 Saturates – 1.5 Sugar – 5 Salt – 0.3 Portion – 100	Fat – Green Saturates – Green Sugar – Green Salt – Green Portion – 100	Fat – Green Saturates – Green Sugar – Green Salt – Green Portion – 100	Here I have tested the upper bounds of the IF statement regarding the green classification. My expected outputs match what I expected so I know that my IF statement is working for green.

Brandon Tonge
ID – 201358937

Fat – 20 Saturates – 5 Sugar – 12.5 Salt – 1.5 Portion – 100	Fat – Amber Saturates – Amber Sugar – Amber Salt – Amber Portion – 100	Fat – Amber Saturates – Amber Sugar – Amber Salt – Amber Portion – 100	Here I have tested the upper bounds of the IF statement regarding the amber classification. My expected outputs match what I expected so I know that my IF statement is working for the upper bounds of amber.
Fat – 19 Saturates – 4 Sugar – 11 Salt – 1 Portion – 100	Fat – Amber Saturates – Amber Sugar – Amber Salt – Amber Portion – 100	Fat – Amber Saturates – Amber Sugar – Amber Salt – Amber Portion – 100	Here I have tested a random number in the amber classification. The outputs are what I expected. Knowing this along with the upper bounds I can assume that my amber IF statement works as I expect it too. I do not need to check the lower bounds as this overlaps with the green IF statement which I have confirmed to work.
Fat – 20.5 Saturates – 6 Sugar – 13 Salt – 2 Portion –10	Fat – Red Saturates – Red Sugar – Red Salt – Red Portion – 100	Fat – Red Saturates – Red Sugar – Red Salt – Red Portion – 100	Here I have tested the ELSE part of each IF statement. I know the upper limit of amber is covered so anything over this should produce red. The actual output matched the expected output, so no corrective action was needed.
Fat – 12 Saturates – 4 Sugar – 10 Salt – 2 Portion – 200	Fat – Red (Portion) Saturates – Red (Portion) Sugar – Red (Portion) Salt – Red (Portion) Portion – 200	Fat – Red (Portion) Saturates – Red (Portion) Sugar – Red (Portion) Salt – Red (Portion) Portion – 200	Here I have tested the portion IF statements. For each on I have set the value just over half the amount needed to trigger the statements and then used to portion size to take me over the threshold. This allows me to make sure the equation works by testing each part of it. From my results I can see that the equations work just as it should and over writes the classification of each value once a certain value has been reached.
Fat – 10.5 Saturates – 3 Sugar – 7.5 Salt – 1.2 Portion – 200	Fat – Amber Saturates – Amber Sugar – Amber Salt – Amber Portion – 200	Fat – Amber Saturates – Amber Sugar – Amber Salt – Amber Portion – 200	Here I am testing to make sure that if I enter the exact number used in the IF statement it does not trigger it. From my results I can confirm that the statement uses a greater then and not an equal too.

Pseudocode –

FUNCTION Selection

OUTPUT “Enter the values”

OUTPUT “Fat value”
INPUT User answer
STORE The variable “fat”

OUTPUT “Saturates value”
INPUT User answer

Brandon Tonge
ID – 201358937

STORE The variable “saturates”

OUTPUT “Sugar value”
INPUT User answer
STORE The variable “sugar”

OUTPUT “Salt value”
INPUT User answer
STORE The variable “salt”

OUTPUT “Portion value”
INPUT User answer
STORE The variable “portion”

IF “fat” is less than or equal to 3
STORE Green in “fat_class” variable
If “fat” is more than 3 and less than or equal to 20
STORE Amber in “fat_class” variable
IF “fat” is more than 20
STORE Red in “fat_class” variable

IF “saturates” is less than or equal to 1.5
STORE Green in “saturates_class” variable
If “saturates” is more than 1.5 and less than or equal to 5
STORE Amber in “saturates_class” variable
IF “saturates” is more than 5
STORE Red in “saturates_class” variable

IF “sugar” is less than or equal to 5
STORE Green in “sugar_class” variable
If “sugar” is more than 5 and less than or equal to 12.5
STORE Amber in “sugar_class” variable
IF “sugar” is more than 12.5
STORE Red in “sugar_class” variable

IF “salt” is less than or equal to 0.3
STORE Green in “salt_class” variable
If “salt” is more than 0.3 and less than or equal to 1.5
STORE Amber in “salt_class” variable
IF “salt” is more than 1.5
STORE Red in “salt_class” variable

IF “fat” * (“portion / 100”) is more than 21
STORE Red (portion) in “fat_class”

IF “saturates” * (“portion / 100”) is more than 6
STORE Red (portion) in “saturates_class”

IF “sugar” * (“portion / 100”) is more than 15
STORE Red (portion) in “sugar_class”

Brandon Tonge
ID – 201358937

IF “salt” * (“portion / 100”) is more than 2.4
STORE Red (portion) in “salt_class”

OUTPUT “portion” variable
OUTPUT “fat_class” and “fat” variables
OUTPUT “saturates_class” and “saturates” variables
OUTPUT “sugar_class” and “sugar” variables
OUTPUT “salt_class” and “salt” variables