Assessment Task 1.1 Functional Programming (Milestone Exercise 1)

RAFFY BASAL

Objectives

This activity aims to assess the student's ability to implement fundamental Python programming and fundamental programming paradigms including controls, loops, and functions.

Instructions:

Perform the following tasks.

Submit the pdf of your accomplished work.

Problem Items

1. Create a dictionary to contain a pair of values, an ID number and a list containing user details, such as name, height, weight and BMI.

2. Create functions that can perform the following tasks:

2a) Add an element to the dictionary with its new ID# and aforementioned details,

2b) Print a specific user ID, and

2c) Print all details contained.

1. Consider the following edge cases and show how you're able to solve them in your code:

3a) What if the new values to add to the dictionary has duplicate values?

3b) How are you handling empty values that the user inputs?

3c) Are you able to handle cases wherein height or weight have non-numeric values?

3d) How do you handle the situation where the ID# for the print user function is not passed a valid ID?

```
# Create a dictionary to contain a pair of values, an ID number and a list containing user details, such as name, height, weight and BMI.

Dict = {"ID": 1,
        "name": "Raffy Basal",
        "height": 1.65,  # in meters
        "weight": 65,     # in kg
        "BMI": 65 / (1.65 ** 2)  # weight in kilograms (kg) divided by
height in meters (m) squared
    }

Dict[2] = [2, ("Bethsy Honda", 1.40, 45, 45 / (1.40 ** 2))]

print(Dict)
print(Dict[2])
```

```
{'ID': 1, 'name': 'Raffy Basal', 'height': 1.65, 'weight': 65, 'BMI':
23.875114784205696, 2: [2, ('Bethsy Honda', 1.4, 45,
22.95918367346939)]}
[2, ('Bethsy Honda', 1.4, 45, 22.95918367346939)]
```

```
# 2. Create functions that can perform the following tasks:

# a new empty dictionary data is created, allow to store key-value
pairs
dict_data = {}

## 2a) Add an element to the dictionary with its new ID# and
aforementioned details

# definening a function that is unique and can't add if the one or
more value is missing
def add_user(user_id, name, height, weight):
    bmi = weight / (height ** 2)   # weight in kilograms (kg) divided
by height in meters (m) squared

    dict_data[user_id] = {
        "name": name,
        "height": height,
        "weight": weight,
        "BMI": bmi # calling the function BMI, w/c is define in the
previous code
    }

def print_user(user_id):
    if user_id in dict_data:
        print({user_id: dict_data[user_id]})
    else:
        print("User ID not found.")
# if user id is not defined

def print_all_users():
    for user_id, details in dict_data.items():
        print({user_id: details})
# print data in a latest dictionary
```

```
## 2a) Add an element to the dictionary with its new ID# and aforementioned details,
add_user(1, "Raffy Toyota", 1.65, 65)
add_user(2, "Bethsy Honda", 1.40, 45)
print(dict_data)

## 2b) Print a specific user ID,
# calling the define functuion of printing the data using the specific user_id
print_user(2)
print_user(3)


## 2c) Print all details contained.
# calling the define fucntion of printing all the data in the latest dictionary
print_all_users()

print_user(3)
del dict_data [2] # deleting data from dictionary
print_all_users()
```

```
{1: {'name': 'Raffy Toyota', 'height': 1.65, 'weight': 65, 'BMI': 23.875114784205696}, 2: {'name': 'Bethsy Honda', 'height': 1.4, 'weight': 45, 'BMI': 22.95918367346939}}
{2: {'name': 'Bethsy Honda', 'height': 1.4, 'weight': 45, 'BMI': 22.95918367346939}}
User ID not found.
{1: {'name': 'Raffy Toyota', 'height': 1.65, 'weight': 65, 'BMI': 23.875114784205696}}
{2: {'name': 'Bethsy Honda', 'height': 1.4, 'weight': 45, 'BMI': 22.95918367346939}}
User ID not found.
{1: {'name': 'Raffy Toyota', 'height': 1.65, 'weight': 65, 'BMI': 23.875114784205696}}
```

```
# 3. Consider the following edge cases and show how you're able to solve them in your code:

## 3d) How do you handle the situation where the ID# for the print user function is not passed a valid ID?
def main():

    dict_data = {}

    def is_numeric(value): # defining the value type of the inputed data
        return isinstance(value, (int, float)) and value >= 0  # Check if value is int or float and non-negative
    # definening a function that is unique and can't add if the one or
more value is missing
    def add_user(user_id, name, height, weight):
        if user_id in dict_data: # checking if the dictionary key are
used Twice, it will not proceed if duplicated
            print("Duplicate ID. User ID already exists.")
            return

        if not name or not is_numeric(height) or not
is_numeric(weight):
            print("Invalid input. Please provide valid name, height,
and weight.")
            return

        # definening a function that is unique and can't add if the
one or more value is missing
        bmi = weight / (height ** 2)   # weight in kilograms (kg)
divided by height in meters (m) squared
        dict_data[user_id] = {
            "name": name,
            "height": height,
            "weight": weight,
            "BMI": bmi # calling the function BMI, w/c is define in
the previous code
        }

    def print_user(user_id):
        if user_id in dict_data:
            print({user_id: dict_data[user_id]})
        else:
            print("User ID not found.")
    # if user id is not defined

    def print_all_users():
        for user_id, details in dict_data.items():
            print({user_id: details})
# print data in a latest dictionary

main()

## 3a) What if the new values to add to the dictionary has duplicate
values?

add_user(1, "Raffy Toyota", 1.65, 65) # adding valid id data
add_user(2, "Bethsy Honda", 1.40, 45) # adding valid id data
add_user(1, "Raffy Avanza", 1.65, 65) # adding duplicate user ID
add_user(3, "Raffy Toyota", 1.65, 65) # adding unique user ID but same
user details
print_all_users()
```

```
## 3b) How are you handling empty values that the user inputs?

add_user(4, "", 1.65, 65) # adding unique user ID but missing one details

## 3c) Are you able to handle cases wherein height or weight have non-numeric values?

add_user(4, "Raffy Toyota", "1.65", 65)
add_user(4, "Raffy Toyota", 1.65, -65)
# adding unique user ID but the one of the details are in str type and negative value
# it was previously defined in the isnumeric above it has to be int or float and must be positive

## 3d) How do you handle the situation where the ID# for the print user function is not passed a valid ID?

print_user(1)
print_user(2)
print_user(3)
print_user(4)
print_all_users()

# it will return a "User ID not found." if the user id inputed is not a valid id.
```

```
Duplicate ID. User ID already exists.
Duplicate ID. User ID already exists.
Duplicate ID. User ID already exists.
Duplicate ID. User ID already exists.
{1: {'name': 'Raffy Toyota', 'height': 1.65, 'weight': 65, 'BMI': 23.875114784205696}}
{2: {'name': 'Bethsy Honda', 'height': 1.4, 'weight': 45, 'BMI': 22.95918367346939}}
{3: {'name': 'Raffy Toyota', 'height': 1.65, 'weight': 65, 'BMI': 23.875114784205696}}
Invalid input. Please provide valid name, height, and weight.
Invalid input. Please provide valid name, height, and weight.
Invalid input. Please provide valid name, height, and weight.
{1: {'name': 'Raffy Toyota', 'height': 1.65, 'weight': 65, 'BMI': 23.875114784205696}}
{2: {'name': 'Bethsy Honda', 'height': 1.4, 'weight': 45, 'BMI': 22.95918367346939}}
{3: {'name': 'Raffy Toyota', 'height': 1.65, 'weight': 65, 'BMI': 23.875114784205696}}
User ID not found.
{1: {'name': 'Raffy Toyota', 'height': 1.65, 'weight': 65, 'BMI':
```

```
23.875114784205696}}
{2: {'name': 'Bethsy Honda', 'height': 1.4, 'weight': 45, 'BMI':
22.95918367346939}}
{3: {'name': 'Raffy Toyota', 'height': 1.65, 'weight': 65, 'BMI':
23.875114784205696}}
```