### **"My Personal Dictionary"**

#### **Objective:**

To develop a Python program named "My Personal Dictionary" that incorporates a wide range of Python programming skills you've learned so far.

This program will serve as a digital dictionary where users can add, search, and view words and their meanings.

#### **Skills Applied:**

* Basic Python Syntax and Variables
* Data Structures: Lists, Tuples, Dictionaries
* Functions (including Lambda)
* Conditional Statements and Loops
* Basic Object-Oriented Programming

#### **Detailed Assignment Instructions:**

* **Program Initialization (Week 2 Skills)**
  + Start the program with a **greeting function** that asks for the user's name and provides a personalized welcome message.
  + Use variables to store the user’s name and any other *necessary information*.
* **Dictionary Data Structure (Week 3 Skills)**
  + Create a **dictionary** to store **words** (as **keys**) and their **meanings** (as **values**).
  + Optionally, use a list to keep track of the recently added words and a tuple for categories of words (like 'Science', 'Math', 'Literature')
* **User Interaction and Menu (Week 8 Skills)**
  + Develop a simple text-based **menu system** using conditional statements (if, elif, else) that allows users to choose between adding a word, searching for a word, and viewing all words.
  + Use a loop (while or for) to keep the program running until the user decides to exit.
* **Adding and Searching Words (Week 5 Skills)**
  + Implement **functions** for adding a new word to the dictionary and for searching for a word.
  + When adding words, **check** if the word already exists and **handle** duplicates appropriately.
* **Display All Words (Week 6-7 Skills)**
  + Use a **function** to display all words in the dictionary.
    - Consider formatting the output for better readability.
  + Use a **lambda function** to sortthe words alphabetically before displaying them.
* **Basic Object-Oriented Programming (Week 9-10 Skills)**
  + Encapsulate the dictionary functionalities within a class. The class should have methods for adding words, searching words, and displaying all words.
  + Ensure proper use of the \_\_init\_\_ method for class initialization.
* **Documentation and Comments (Ongoing)**
  + Throughout the program, add comments explaining the functionality of different sections of the code.
  + Include a program header comment with the title of the program, your name, date, and a brief description of the program.
* **Error Handling (Week 8 Skills)**
  + Implement error handling to **manage incorrect inputs** or **unexpected scenarios (like searching for a non-existent word)**.

#### **Example of the Program Menu (Your Menu must be customized with your own choices):**

****'''

Welcome to My Personal Dictionary, [User's Name]!

--- Main Menu ---

1: Add a word

2: Search a word

3: Display all words

4: Exit

Enter your choice:

'''



#### **Assessment Rubric (Out of 50 Marks):**

* **Functionality (20 marks)**: All features work as expected.
* **Code Organization and Style (10 marks)**: Clear structure, readability, and adherence to Python conventions.
* **Documentation and Comments (10 marks)**: Comprehensive, clear, and helpful comments.
* **Error Handling (5 marks)**: Effective handling of errors and edge cases.
* **Creative Use of Concepts (5 marks)**: Innovative and efficient application of the concepts taught.

#### **Note:**

* Be creative! Think about how you can enhance the dictionary's features and user experience. Select a theme if you want to, like Gaming, Banking, or anything else.
* Maybe keep some code in a separate custom module?
* Focus on writing clean, readable, and well-documented code.
* Test your program thoroughly to ensure all functionalities work as expected.

----------------

**Submission:**

* Create a Python script (.py file) containing your code.
* Ensure your script is well-documented with comments explaining the code. It has to have your name in the comments (include student# - if there is one).
* Submit your Python script file (.py) on DC-CONNECT, following the guidelines provided by your instructor.

-----------------

**Demo Code (to guide you, your code must look a bit more enhanced and personalized):**

# My Personal Dictionary

class Dictionary:

def \_\_init\_\_(self):

# TODO: Initialize a data structure to store the dictionary entries

pass

def add\_word(self, word, meaning):

# TODO: Implement the logic to add words and their meanings

pass

def search\_word(self, word):

# TODO: Implement the logic to search for a word and return its meaning

pass

def display\_all(self):

# TODO: Implement the logic to display all words and meanings

pass

def greet\_user():

# TODO: Ask for the user's name and greet them

pass

def main\_menu():

my\_dict = Dictionary()

while True:

# TODO: Create a simple menu with options to add, search, and display words

print("\n--- Main Menu ---")

print("1: Add a word")

print("2: Search a word")

print("3: Display all words")

print("4: Exit")

choice = input("Enter your choice: ")

# TODO: Implement the logic for menu options

if choice == '1':

pass # Add logic to input word and meaning, and call add\_word

elif choice == '2':

pass # Add logic to input word and call search\_word

elif choice == '3':

pass # Add logic to call display\_all

elif choice == '4':

print("Exiting the dictionary. Goodbye!")

break

else:

print("Invalid choice. Please try again.")

# Start of the program

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

greet\_user()

main\_menu()

Assignment Challenges:

using dictionaries - add, remove elements?