

---

## Data Structure (Lists, Tuples, and Dictionaries)

---

### Important Instruction for Week 02 Lab:

Do not use AI tools or external code generators to complete these exercises. They are intended for you to practice your own coding abilities. Think through the logic, reference Python syntax, and work through the challenges independently.

---

### Your task is to Write a Python Programming Language to complete the exercises below:

1. Write a function that takes a list and creates a new list with the elements in reverse order.

Expected input/output:

Input: [1, 2, 3, 4, 5]

Output: [5, 4, 3, 2, 1]

2. Create a list of squares of even numbers from 1 to 20 using list comprehension.

Expected input/output:

Output: [4, 16, 36, 64, 100, 144, 196, 256, 324, 400]

3. Write a function that takes two lists and merges them into a single list, ensuring that each element appears only once.

Expected input/output:

Input: [1, 2, 3], [2, 3, 4, 5]

Output: [1, 2, 3, 4, 5]

4. Write a function that takes a tuple of numbers and returns a new tuple containing the maximum and minimum values from the input tuple.

Expected input/output:

Input: (10, 5, 8, 12, 3)

Output: (12, 3)

5. By using a tuple, create a list of cities with their names and GPS coordinates, then write a function to display the city name and its corresponding coordinates.

Expected input/output:

Output:

City: Phnom Penh, Latitude: 11.5564, Longitude: 104.9282

City: Siem Reap, Latitude: 13.3622, Longitude: 103.8597

City: Battambang, Latitude: 13.0957, Longitude: 103.2022

6. You have a dictionary where the keys are numbers and the values are their current values.

Use a **lambda function** to double each value in the dictionary.

Expected input/output:

Input: {1: 10, 2: 20, 3: 30, 4: 40}

Output: {1: 20, 2: 40, 3: 60, 4: 80}

7. Write a function that takes a string as input and returns a dictionary with the frequency of each character in the string.

Expected input/output:

Input: "hello"

Output: {'h': 1, 'e': 1, 'l': 2, 'o': 1}

8. Write a function that takes two dictionaries and merges them into one. If there are duplicate keys, the values should be added together (assuming they are numbers).

Expected input/output:

Input: {'a': 1, 'b': 2}, {'b': 3, 'c': 4}

Output: {'a': 1, 'b': 5, 'c': 4}

## (Bonus) Data Management

Use Python to manage a list of user accounts, each represented as a dictionary. Practice methods for lists and dictionaries, including adding, removing (1 user), updating, and deleting entries.

### Instructions:

1. Create a list of dictionaries where each dictionary represents a user account with the following keys:

- username: A string representing the user's name.
- email: A string representing the user's email address.
- status: A string representing the account status (e.g., 'active', 'suspended').

Example:

```
users = [  
    {"username": "alice", "email": "alice@example.com", "status": "active"},  
    {"username": "bob", "email": "bob@example.com", "status": "suspended"},  
    {"username": "charlie", "email": "charlie@example.com", "status": "active"}  
]
```

2. Implement the following operations:

- Add a new user: Write a function `add_user(users, username, email, status)` that adds a new user to the list.
- Remove a user: Write a function `remove_user(users, username)` that removes a user based on the username.
- Update user status: Write a function `update_user_status(users, username, new_status)` that updates the status of a user.
- Display all users: Write a function `display_users(users)` that prints out all users in a formatted way.

3. (Optional)

- Count active users: Write a function `count_active_users(users)` that returns the number of active users.

**What you need to call and show:**

```
add_user(users, "dave", "dave@example.com", "active")
remove_user(users, "bob")
update_user_status(users, "charlie", "suspended")
display_users(users)
print("Active users:", count_active_users(users))
```

**Expected Output:**

User 'dave' added successfully.  
User 'bob' removed successfully.  
User 'charlie' status updated to 'suspended'.

Current user accounts:

Username	Email	Status
-----		
alice	alice@example.com	active
charlie	charlie@example.com	suspended
dave	dave@example.com	active

Total active users: 2

***Reminder: Do not use AI tools to generate your solution. Think through how you would logically check each condition and write your code accordingly.***



At the end of this lab, you need to produce one file named: **full\_name\_W02\_Lab.py**

- Example: Chan\_Dara\_W02\_Lab.py

🚨 **Note:** If you cannot submit the file in .py format, please submit it in a .zip format with the same file name (e.g., Chan\_Dara\_W02\_Lab.zip).