## **Day 2: Lists, Tuples, Dictionaries, Sets, Classes, and Functions**

### **Lists and Tuples**

#### **Lists**

* **Topics:**
  + Creating lists, indexing, slicing, iterating.
  + List methods: append(), extend(), insert(), remove(), pop(), sort(), reverse().
* **Activities:**
  + Write programs using list methods:
    - Add/remove elements dynamically.
    - Find the second largest number in a list.
    - Combine two lists and remove duplicates.

#### **Tuples**

* **Topics:**
  + Creating tuples, unpacking, immutability.
  + Tuple operations: indexing, slicing, count(), index().
* **Activities:**
  + Convert a list to a tuple and vice versa.
  + Create a tuple of numbers and find the sum, max, and min.
  + Use tuple unpacking to swap values of two variables.

### **Dictionaries and Sets**

#### **Dictionaries**

* **Topics:**
  + Creating dictionaries, accessing keys and values.
  + Dictionary methods: get(), keys(), values(), items(), pop(), update().
* **Activities:**
  + Write programs to:
    - Create a dictionary where keys are numbers from 1 to 5 and values are their cubes.
    - Merge two dictionaries.
    - Count occurrences of each word in a string and store the results in a dictionary.

#### **Sets**

* **Topics:**
  + Creating sets, adding and removing elements.
  + Set operations: union, intersection, difference, symmetric difference.
* **Activities:**
  + Write programs to:
    - Find unique elements from two lists using sets.
    - Check if one set is a subset of another.
    - Use set comprehensions to generate squares of even numbers from 1 to 20.

### **Classes and Functions**

#### **Functions**

* **Topics:**
  + Defining functions (def keyword), parameters, return values.
  + Default and keyword arguments.
  + Lambda functions.
* **Activities:**
  + Write a function to calculate the factorial of a number.
  + Create a function to find the largest element in a list without using the max() function.
  + Write a lambda function to filter even numbers from a list.

#### **Classes**

* **Topics:**
  + Creating classes and objects.
  + Attributes, methods, and \_\_init\_\_().
  + Class vs. instance attributes.
  + Understanding self.
* **Activities:**
  + Create a class to represent a BankAccount with methods to deposit, withdraw, and check balance.
  + Implement a class to represent a Rectangle with methods to calculate area and perimeter.
  + Create a Student class with attributes for name, age, and grades. Add methods to calculate average grade and display student details.

### **Hands-on Assignment**

**Assignment 2:**

1. **List Operations:**
   * Write a program that:
     + Accepts a list of integers.
     + Returns a new list with duplicates removed, sorted in ascending order.
2. **Tuple Manipulations:**
   * Write a program that:
     + Creates a tuple of numbers from 1 to 10.
     + Finds the sum and product of all numbers in the tuple.
3. **Dictionary Operations:**
   * Write a program to:
     + Count occurrences of each character in a given string using a dictionary.
     + Delete a specific key from the dictionary and handle the case where the key does not exist.
4. **Set Operations:**
   * Write a program that:
     + Accepts two lists of numbers.
     + Converts them to sets and finds their union, intersection, and difference.
5. **Class-Based Assignment:**
   * Create a Library class to manage a collection of books. Include methods to:
     + Add a book.
     + Remove a book.
     + Display all books.