Cycle 1

Set 1

An example of a Python program that demonstrates various operations and manipulations with Lists:

# Creating a list

fruits = ["apple", "banana", "cherry", "date"]

# Accessing elements

print("First fruit:", fruits[0])

print("Last fruit:", fruits[-1])

print("Slice of fruits:", fruits[1:3])

# Modifying elements

fruits[1] = "grape"

print("Modified fruits:", fruits)

# Adding elements

fruits.append("elderberry")

print("After adding:", fruits)

# Removing elements

removed\_fruit = fruits.pop(2)

print("Removed fruit:", removed\_fruit)

print("Remaining fruits:", fruits)

# Length of the list

print("Number of fruits:", len(fruits))

# Sorting the list

fruits.sort()

print("Sorted fruits:", fruits)

# Reversing the list

fruits.reverse()

print("Reversed fruits:", fruits)

# Copying a list

new\_fruits = fruits.copy()

print("Copied list:", new\_fruits)

# Clearing the list

fruits.clear()

print("Cleared fruits:", fruits)

An example of a Python program that demonstrates various operations and manipulations with dictionaries:

# Creating a dictionary

student = {

"name": "John",

"age": 20,

"major": "Computer Science",

"gpa": 3.5

}

# Accessing values

print("Student name:", student["name"])

print("Student age:", student.get("age"))

# Modifying values

student["gpa"] = 3.8

print("Modified GPA:", student["gpa"])

# Adding new key-value pairs

student["university"] = "XYZ University"

print("After adding university:", student)

# Removing a key-value pair

removed\_major = student.pop("major")

print("Removed major:", removed\_major)

print("Remaining student data:", student)

# Checking if a key exists

if "age" in student:

print("Age exists in student dictionary")

# Length of the dictionary

print("Number of key-value pairs:", len(student))

# Accessing keys and values

print("Keys:", student.keys())

print("Values:", student.values())

# Iterating over key-value pairs

for keys, values in student.items():

print(keys, ":", values)

**# Copying a dictionary**

new\_student = student.copy()

print("Copied dictionary:", new\_student)

# Clearing the dictionary

student.clear()

print("Cleared student dictionary:", student)

An example of a Python program that demonstrates various operations and manipulations with tuples:

# Creating a tuple

fruits = ("apple", "banana", "cherry", "date")

# Accessing elements

print("First fruit:", fruits[0])

print("Last fruit:", fruits[-1])

print("Slice of fruits:", fruits[1:3])

# Modifying a tuple (not allowed)

# fruits[1] = "grape" # This will result in an error

# Length of the tuple

print("Number of fruits:", len(fruits))

­

# Checking if an element exists

if "apple" in fruits:

print("Apple exists in the tuple")

# Counting occurrences of an element

print("Occurrences of 'cherry':", fruits.count("cherry"))

# Finding the index of an element

print("Index of 'banana':", fruits.index("banana"))

# Converting a tuple to a list

fruits\_list = list(fruits)

print("Converted to list:", fruits\_list)

# Modifying the list

fruits\_list[1] = "grape"

print("Modified list:", fruits\_list)

# Converting the list back to a tuple

fruits = tuple(fruits\_list)

print("Converted back to tuple:", fruits)

An example of a Python program that demonstrates various operations and manipulations with dictionaries:

# Creating a set

fruits = {"apple", "banana", "cherry", "date"}

# Adding elements to a set

fruits.add("elderberry")

print("After adding:", fruits)

# Removing an element from a set

fruits.remove("cherry")

print("After removing:", fruits)

# Checking if an element exists in a set

if "banana" in fruits:

print("Banana exists in the set")

# Length of a set

print("Number of fruits:", len(fruits))

# Iterating over a set

print("Fruits:")

for fruit in fruits:

print(fruit)

# Set operations

set1 = {1, 2, 3, 4, 5}

set2 = {4, 5, 6, 7, 8}

# Union of sets

union\_set = set1.union(set2)

print("Union set:", union\_set)

# Intersection of sets

intersection\_set = set1.intersection(set2)

print("Intersection set:", intersection\_set)

# Difference between sets

difference\_set = set1.difference(set2)

print("Difference set:", difference\_set)

# Checking for subset or superset

is\_subset = set1.issubset(set2)

is\_superset = set1.issuperset(set2)

print("Is set1 a subset of set2:", is\_subset)

print("Is set1 a superset of set2:", is\_superset)

# Clearing a set

fruits.clear()

print("Cleared set:", fruits)

**Set 2**

I Phonebook Application: Create a phonebook application that allows users to store and retrieve contact information using dictionaries. The program should have the following functionalities:

* Add a new contact (name and phone number)
* Update an existing contact
* Delete a contact
* Search for a contact by name
* Display all contacts

II

1 Student Grades: Write a program that allows users to enter the grades of students and calculates the average grade using a list. The program should have the following functionalities:

* Add a new grade to the list
* Calculate and display the average grade
* Display all grades

2 Number Sorting:Write a program that reads a list of numbers from the user and sorts them in ascending order using a list. The program should display the sorted list.

1. List Operations: Write a program that performs various operations on a list, such as

adding elements,

removing elements,

searching for an element,

and displaying the list.

The program should allow users to interactively perform these operations.

2 List Statistics: Develop a program that reads a list of numbers from the user and calculates various statistics using a list. The program should display the minimum, maximum, sum, and average of the numbers

III

1 Create a tuple containing the names of five countries. Print the tuple.

1. Access the third country in the tuple and print its name.
2. Iterate over the tuple and print each country name on a separate line.
3. Check if "Germany" is present in the tuple. Print a suitable message based on the result.
4. Concatenate the tuple with another tuple containing two additional countries. Print the combined tuple.
5. Find the length of the updated tuple and print the result.
6. Find the index of "France" in the tuple and print its position.
7. Count the number of occurrences of "United States" in the tuple and print the count.
8. Convert the tuple to a list and print the list.
9. Sort the list in alphabetical order and print the sorted list.
10. Convert the sorted list back to a tuple and print the tuple.

IV

Create a set containing the names of five countries. Print the set.

1. Check if "Germany" is present in the set. Print a suitable message based on the result.
2. Add two new countries to the set. Print the updated set.
3. Remove one country from the set. Print the updated set.
4. Create another set containing the names of three countries. Print the second set.
5. Find the intersection of the two sets and print the common countries.
6. Find the union of the two sets and print all the unique countries.
7. Check if the first set is a subset of the second set. Print the result.
8. Create a new set that includes all the countries from both sets. Print the merged set.
9. Convert the set to a sorted list and print the sorted list.
10. *Clear the set and print the empty set.*