PES UNIVERSITY, BENGALURU

Perseverance | Excellence | Service

UE25CS151A – PYTHON FOR COMPUTATIONAL PROBLEM SOLVING LAB MANUAL

WEEK 6

TOPICS:

Programs on Sets,Dictionaries and Strings

OBEJCTIVE:

Solve practical problems using set, dictionary and string to manage collections of data.

Problem Statement 1: (set)

In a school event, two groups of students participate in different activities.

- List1 represents students (by ID numbers) in the **music club**, and
- List2 represents students (by ID numbers) in the **sports club.**
- Some students participate in **both**.

Write a Python program to convert these lists to sets and perform the following operations:

- Find all students participating in at least one club.
- Find students participating in both clubs.
- Find students only in the music club.
- Find students who are in exactly one club.

Display the results clearly, showing the original lists and the results of each operation.

Sample Output:

```
Music club student IDs: [1, 3, 5, 4, 7, 2, 9]
Sports club student IDs: [3, 4, 5, 6, 7]
All students in at least one club: {1, 2, 3, 4, 5, 6, 7, 9}
Students in both clubs: {3, 4, 5, 7}
Students only in music club: {1, 2, 9}
Students in exactly one club: {1, 2, 6, 9}
```

Problem Statement 2: (set)

Write a Python program to count how many stones are jewels. You are given a string 'jewels' representing the types of stones that are jewels, and a string ‘stones’ representing the stones you have. Each character in 'stones' is a type of stone, and you need to count how many of these are present in 'jewels'. Take the strings as user input.

Sample output:

Jewels: aAb
Stones: aAAbcccadddaA
Number of stones that are jewels: 7

Problem Statement 3: (Dictionary)

Write a Python program to store student names and their grades in a dictionary. Compute the average grade and identify students with grades above the average.

Sample output:

```
Student grades: {'Ram': 85, 'Sita': 90, 'Akbar': 78, 'Antony': 92}
Average grade: 86.25
Students above average: {'Sita': 90, 'Antony': 92}
```

Problem Statement 4: (Dictionary)

Python Program for Detecting Duplicates with Counts and Non-Duplicates

In a voting system, each number in a list represents a candidate ID voted by a student.

- Write a Python program to analyse the votes in the given list.
- Determine if there are any duplicate votes (same candidate ID appearing more than once), create a dictionary showing the candidate IDs that received multiple votes along with their vote counts, and
- Identify the candidate IDs that received exactly one vote, storing them in a set.
- Display the original list, whether duplicates exist, the dictionary of duplicated candidate IDs with their counts, and the set of non-duplicated candidate IDs.

Sample output:

```
Vote list (candidate IDs): [1, 2, 3, 1, 1, 2, 3, 4, 5, 6, 7]
Contains duplicate votes? True
Duplicated candidate IDs with vote counts: {1: 3, 2: 2, 3: 2}
Non-duplicated candidate IDs: {4, 5, 6, 7}
```

Problem Statement 5: (Strings)

Write a python program to count the number of vowels and consonants in the string.

Sample output:

```
String: Python Programming
Vowel count: 4
Consonant count: 13
```

Problem Statement 6: (Strings)

Write a Python program that takes an input string, replaces every occurrence of the letter 'o' with the digit '0', and then converts the entire modified string to uppercase.

Sample Output:
Original string: Hello world, how are you today?
Output: HELL0 W0RLD, H0W ARE Y0U T0DAY?

If the implementation is hard to explain, it's a bad idea – Make it simple