***Q1. What are the characteristics of the tuples? Is tuple immutable?***

Tuples use parentheses: To create a tuple, you enclose its elements in parentheses ( ) and separate them with commas.

Tuples are immutable: Once a tuple is created, you cannot add, remove, or modify any elements within it.

Tuples can contain any data type: You can create a tuple containing elements of different data types, such as integers, strings, and floats.

Tuples can be used as keys in dictionaries: Because tuples are immutable, they can be used as keys in dictionaries.

Yes Tuple is immutable.

Q2. What are the two tuple methods in python? Give an example of each method. Give a reason why tuples have only two in-built methods as compared to lists.

The Two methods in tuple are count() and index().

Count() method: This method returns the number of times a specified element appears in a tuple.

Example:

my\_tuple = (1, 2, 2, 3, 4, 2)

count = my\_tuple.count(3)

print(count)

# Output: 1

Index() method(): This method returns the index of the first occurrence of a specified element in a tuple.

Example:

my\_tuple = (1, 2, 2, 3, 4, 2)

index = my\_tuple.index(4)

print(index)

# Output: 4

**Tuples have only two in-built methods compared to lists because tuples are immutable, meaning they cannot be modified once created. As a result, there is no need for methods such as append(), insert(), remove(), or pop() that modify the contents of a tuple. The two methods available for tuples are focused on accessing data within the tuple, rather than modifying the tuple itself.**

Q3. Which collection datatypes in python do not allow duplicate items? Write a code using a set to remove duplicates from the given list.

The set is the collection data Types in Python which do not allow duplicate items.

my\_list = [1, 2, 2, 3, 4, 4, 5, 5]

my\_set = set(my\_list)

new\_list = list(my\_set)

print(new\_list)

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

Q4. Explain the difference between the union() and update() methods for a set.Give an example of each method.

**The main difference between union() and update() methods is that union() returns a new set containing all the unique elements from both sets, whereas update() modifies the original set and adds all the elements from another set into it.**

**union() method: This method returns a new set that contains all the unique elements from both the sets.**

**Example:**

**set1 = {1, 2, 3}**

**set2 = {3, 4, 5}**

**new\_set = set1.union(set2)**

**print(new\_set)**

**# Output: {1, 2, 3, 4, 5}**

**update() method: This method modifies the original set and adds all the elements from another set into it.**

**Example:-**

**set1 = {1, 2, 3}**

**set2 = {3, 4, 5}**

**set1.update(set2)**

**print(set1)**

**# Output: {1, 2, 3, 4, 5}**

**Q5. What is a Dictionary?Give an example.Also,state whether a dictionary is ordered or unordered.**

**Dictionary is also known as an associative array or a hash map.**

**Each element in a dictionary consists of a key and its corresponding value. The key is used to access the value associated with it.**

**Example:**

**my\_dict = {"mango": 2, "pear": 3, "sweetlime": 1}**

**print(my\_dict)**

**# Output: {'mango': 2, 'pear': 3, 'sweetlime': 1}**

**The dictionary in Python is an unordered collection of key-value pairs.**

**Q6. Can we create a nested dictionary?Is so,please give an example by creating a simple one-level nested dictionary.**

**Yes, we can create a nested dictionary in Python. A nested dictionary is a dictionary that contains one or more dictionaries as values of its keys.**

**Here's an example of a simple one-level nested dictionary:**

**my\_dict = {"fruit": {"apple": 2, "banana": 3, "orange": 1}}**

**print(my\_dict)**

**# Output: {'fruit': {'apple': 2, 'banana': 3, 'orange': 1}}**

**Q7. Using setdefault() method,create key named topics in the given dictionary and also add the value of the key as this list[‘Python’,’Machine Learning’,’Deep Learning’]**

**dict1={‘language’:’Python’,’course’:’Data Science Masters’}**

**dict1 = {'language': 'Python', 'course': 'Data Science Masters'}**

**dict1.setdefault('topics', ['Python', 'Machine Learning', 'Deep Learning'])**

**print(dict1)**

**Output**

**{'language': 'Python', 'course': 'Data Science Masters', 'topics': ['Python', 'Machine Learning', 'Deep Learning']}**

**Q8. What are the three view objects in dictionaries? Use the three in-built methods in python to display these three view objects for the given dictionary.**

**dict1={‘Sport’:’Cricket’,’Teams’:[‘India’,’Australia’,’England’,SouthAfrica’,’Sri Lanka’,’New Zealand’]}**

**Dict.keys(),dict.values(),dict.items() are the three view objects in dictionaries also this are the three in built methods in python.**

**dict1 = {'Sport': 'Cricket', 'Teams': ['India', 'Australia', 'England', 'SouthAfrica', 'Sri Lanka', 'New Zealand']}**

**# get the view object for the keys of dict1**

**keys\_view = dict1.keys()**

**print("keys view object:", keys\_view)**

**# get the view object for the values of dict1**

**values\_view = dict1.values()**

**print("values view object:", values\_view)**

**# get the view object for the key-value pairs of dict1**

**items\_view = dict1.items()**

**print("items view object:", items\_view)**

**Output:**

**keys view object: dict\_keys(['Sport', 'Teams'])**

**values view object: dict\_values(['Cricket', ['India', 'Australia', 'England', 'SouthAfrica', 'Sri Lanka', 'New Zealand']])**

**items view object: dict\_items([('Sport', 'Cricket'), ('Teams', ['India', 'Australia', 'England', 'SouthAfrica', 'Sri Lanka', 'New Zealand'])])**