**1) What are the characteristics of the tuples? Is tuple immutable?**

Ans- These tuples are an ordered collection of elements of different data types. Furthermore, we represent them by writing the elements inside the parenthesis separated by commas. We can also define tuples as lists that we cannot change. Therefore, we can call them immutable tuples.

Besides the different kind of brackets used to delimit them, the main difference between a tuple and a list is that the tuple object is immutable. Once we've declared the contents of a tuple, we can't modify the contents of that tuple

**2) 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.

Ans- count() Returns the number of times a specified value occurs in a tuple

index() Searches the tuple for a specified value and returns the position of where it was found.

Lists have several built-in methods. A unexpected change or error is more likely to occur in a list. In a tuple, changes and errors don't usually occur because of immutability. The key takeaways are: The key difference between the tuples and lists is that while the tuples are immutable objects the lists are mutable. This means that tuples cannot be changed while the lists can be modified. Tuples are more memory efficient than the lists

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

**List = [1, 1, 1, 2, 1, 3, 1, 4, 2, 1, 2, 2, 2, 3, 2, 4, 3, 1, 3, 2, 3, 3, 3, 4, 4, 1, 4, 2, 4, 3, 4, 4]**

Ans- Set is a collection which is unordered and unindexed. No duplicate members.

test\_list = [1, 1, 1, 2, 1, 3, 1, 4, 2, 1, 2, 2, 2, 3, 2, 4, 3, 1, 3, 2, 3, 3, 3, 4, 4, 1, 4, 2, 4, 3, 4, 4]

print ("The original list is : "

+ str(test\_list))

# using set()

# to remove duplicated

# from list

test\_list = list(set(test\_list))

# printing list after removal

# distorted ordering

print ("The list after removing duplicates : "

+ str(test\_list))

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

## Ans- Union Method :

This method is used to return the union of a set and the set of elements from one or more iterable like string, list, set. This method takes arbitrary number of iterable objects as argument, which means one or more iterable can be passed. If iterable other than set is passed, it first converts the iterable object to set object and then perform the union operation. It returns a newly created set, which contains all the elements ( distinct ) present in all the iterables.

## **update Method :**

This method is used to return the union of a set and the set of elements from one or more iterable like string, list, set. It is very similar to **union()** method, with difference is that where union() method create and return a new set, containing all the elements ( distinct ) present in all the iterables, update() method updates the set on which this method is called with all the distinct elements present in all the iterables.

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

Ans- Dictionary. Dictionaries are used to store data values in key:value pairs. A dictionary is a collection which is ordered\*, changeable and do not allow duplicates. As of Python version 3.7, dictionaries are ordered. In Python 3.6 and earlier, dictionaries are unordered.

**6) Can we create a nested dictionary? If so, please give an example by creating a simple one-level nested dictionary**

Ans- Nesting Dictionary means putting a dictionary inside another dictionary. Nesting is of great use as the kind of information we can model in programs is expanded greatly.

Dict = { 'Dict1': { },

'Dict2': { }}

print("Nested dictionary 1-")

print(Dict)

# Nested dictionary having same keys

Dict = { 'Dict1': {'name': 'Ali', 'age': '19'},

'Dict2': {'name': 'Bob', 'age': '25'}}

print("\nNested dictionary 2-")

print(Dict)

# Nested dictionary of mixed dictionary keys

Dict = { 'Dict1': {1: 'G', 2: 'F', 3: 'G'},

'Dict2': {'Name': 'Geeks', 1: [1, 2]} }

print("\nNested dictionary 3-")

print(Dict)

**7) 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'**

Ans-

dict1 = { 'Python', 'Machine Learning’, 'Deep Learning'}

print("Dictionary before using setdefault():", dict1)

# using setdefault() when key is non-existing

ret\_value = dict1 ('C', " Python ")

print("Return value of setdefault():", ret\_value)

print("Dictionary after using setdefault():", dict1)

**8) 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', 'South Africa', 'Sri Lanka', 'New Zealand']}**

Ans- The main view objects of dictionary in python are keys, values and items.

d = {} # empty dictionary

numNames{'Sport': 'Cricket' , 'Teams': ['India', 'Australia', 'England', 'South Africa', 'Sri Lanka', 'New Zealand']}

# int key, string value

decNames={1.5:"One and Half", 2.5: "Two and Half", 3.5:"Three and Half"} # float key, string value

items={("Parker","Reynolds","Camlin"):"pen", ("LG","Whirlpool","Samsung"): "Refrigerator"} # tuple key, string value

romanNums = {'I':1, 'II':2, 'III':3, 'IV':4, 'V':5} # string key, int value