Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2023-24

(310908) Python Programming Laboratory

****

Class: FY-MCA Shift / Div : F2 / B Roll Number : 51124

Name: Sameer Kakade Assignment No:5 Date ofImplementation:14/10/2023

****

1. Program to create a tuple with different data types.

```
mixed_tuple = (1, "hello", 3.14, True)

print(mixed_tuple)
```

Output:

```
(1, 'hello', 3.14, True)
```

2. Program to add an item in a tuple.

```
original_tuple = (1, 2, 3)

new_tuple = original_tuple + (4,)

print(new_tuple)
```

Output:

```
(1, 2, 3, 4)
```

3. Program to convert a tuple to a string.

```
tuple_of_chars = ('h', 'e', 'l', 'l', 'o')

string = ''.join(tuple_of_chars)
```

```
print(string)
```

Output:

```
hello
```

4. Program to find the repeated items of a tuple.

```
tuple_with_duplicates = (1, 2, 3, 2, 4, 2, 5)

repeated_items = [item for item in tuple_with_duplicates if tuple_with_duplicates.count(item) > 1]

print(set(repeated_items))
```

Output:

```
{2}
```

5. Program to check whether an element exists within a tuple.

```
numbers = (1, 2, 3, 4, 5)

element_to_check = 3

exists = element_to_check in numbers

print(exists)
```

Output:

```
True
```

6. Program to convert a list to a tuple.

```
list_of_numbers = [1, 2, 3, 4, 5]

converted_tuple = tuple(list_of_numbers)

print(converted_tuple)
```

Output:

```
(1, 2, 3, 4, 5)
```

7. Program to remove an item from a tuple.

```
original_tuple = (1, 2, 3, 4, 5)

item_to_remove = 3

new_tuple = tuple(item for item in original_tuple if item != item_to_remove)

print(new_tuple)
```

Output:

```
(1, 2, 4, 5)
```

8. Program to slice a tuple.

```
original_tuple = (1, 2, 3, 4, 5)

sliced_tuple = original_tuple[2:4]

print(sliced_tuple)
```

Output:

```
(3, 4)
```

9. Program to find the index of an item of a tuple.

```
original_tuple = (1, 2, 3, 4, 5)

item_to_find = 3

index = original_tuple.index(item_to_find)

print(index)
```

Output:

```
2
```

10. Program to find the length of a tuple.

```
my_tuple = (1, 2, 3, 4, 5)

length = len(my_tuple)

print(length)
```

Output:

```
5
```

11. Program to convert a tuple to a dictionary.

```
tuple_data = ((1, 'one'), (2, 'two'), (3, 'three'))

dictionary = dict(tuple_data)

print(dictionary)
```

Output:

{1: 'one', 2: 'two', 3: 'three'}

12. Program to unzip a list of tuples into individual lists.

```
list_of_tuples = [(1, 'one'), (2, 'two'), (3, 'three')]

numbers, words = zip(*list_of_tuples)

print(numbers)

print(words)
```

Output:

(1, 2, 3)

('one', 'two', 'three')

13. Program to reverse a tuple.

```
my_tuple = (1, 2, 3, 4, 5)

reversed_tuple = my_tuple[::-1]

print(reversed_tuple)
```

Output:

(5, 4, 3, 2, 1)

14. Program to create a set.

```
my_set = {1, 2, 3, 4, 5}

print(my_set)
```

Output:

```
{1, 2, 3, 4, 5}
```

15. Program to add member(s) in a set.

```
my_set = {1, 2, 3}

my_set.add(4)

print(my_set)
```

Output:

```
{1, 2, 3, 4}
```

16. Program to remove item(s) from a set.

```
my_set = {1, 2, 3, 4, 5}

my_set.remove(3)

print(my_set)
```

Output:

```
{1, 2, 4, 5}
```

17. Program to create an intersection of sets.

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

set2 = {3, 4, 5, 6}

intersection_set = set1.intersection(set2)

print(intersection_set)
```

Output:

{3, 4}

18. Program to create a union of sets.

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

set2 = {3, 4, 5, 6}

union_set = set1.union(set2)

print(union_set)
```

Output:

{1, 2, 3, 4, 5, 6}

19. Program to create set difference.

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

set2 = {3, 4, 5, 6}

difference_set = set1.difference(set2)

print(difference_set)
```

Output:

{1, 2}

20. Program to create a symmetric difference.

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

set2 = {3, 4, 5, 6}

symmetric_difference_set = set1.symmetric_difference(set2)

print(symmetric_difference_set)
```

Output:

{1, 2, 5, 6}

21. Program to check if a set is a subset of another set.

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

set2 = {1, 2, 3, 4, 5}

is_subset = set1.issubset(set2)

print(is_subset)
```

Output:

True

22. Program to create frozensets.

```
my_frozenset = frozenset([1, 2, 3, 4, 5])

print(my_frozenset)
```

Output:

```
frozenset({1, 2, 3, 4, 5})
```

23. Program to find maximum and the minimum value in a set.

```
my_set = {1, 2, 3, 4, 5}

maximum_value = max(my_set)

minimum_value = min(my_set)

print("Maximum:", maximum_value)

print("Minimum:", minimum_value)
```

Output:

```
Maximum: 5

Minimum: 1
```

24. Program to check if a given set is a superset of itself and superset of another given set.

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

is_superset_of_itself = set1.issuperset(set1)

set2 = {1, 2}

is_superset_of_set2 = set1.issuperset(set2)

print("Is superset of itself:", is_superset_of_itself)
```

```
print("Is superset of set2:", is_superset_of_set2)
```

Output:

```
Is superset of itself: True

Is superset of set2: True
```

25. Program to remove the intersection of a 2nd set from the 1st set.

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

set2 = {3, 4, 5, 6}

set1.difference_update(set2)

print(set1)
```

Output:

```
{1, 2}
```