

# Practical Assignment 3

## Basic Set Programs

In [ ]: 1.Create a set and display its element

```
In [3]: my_set = {1,2,3,4,5}
print("Elements of the set :",my_set)
```

Elements of the set : {1, 2, 3, 4, 5}

In [ ]: 2.Add an element to a set

```
In [7]: my_set = {1,2,3}
my_set.add(4)
print(my_set)
my_set.add(2)
print(my_set)
```

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

In [ ]: 3.Remove an element from a set

```
In [19]: my_set = {1,2,3,4,5}
my_set.remove(3)
print(my_set)
```

{1, 2, 4, 5}

4.Clear all elements from a set

```
In [21]: my_set = {"apple","banana","cherry"}
my_set.clear()
print(my_set)
```

set()

In [ ]: 5.Copy a set to another set

```
In [33]: original_set = {1,2,3,4}
copied_set = original_set.copy()
copied_set.add(5)
```

```
print(original_set)
print(copied_set)
```

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

6. Check if an element exists in a set

```
In [37]: print("6. Is 20 in the set?", 20 in my_set)
```

6. Is 20 in the set? False

```
In [ ]: 7. Find the length of set
```

```
In [39]: print("7. Length of the set:", len(my_set))
```

7. Length of the set: 0

```
In [ ]: 8. Iterate through a set using a loop
```

```
In [41]: print("8. Iterating through the set:")
for element in my_set:
    print(element)
```

8. Iterating through the set:

## Set operations

```
In [52]: set_a = {1, 2, 3, 4}
set_b = {3, 4, 5, 6}
union_set = set_a | set_b
print("Union of A and B:", union_set)
```

Union of A and B: {1, 2, 3, 4, 5, 6}

```
In [56]: intersection_set = set_a & set_b
print("Intersection of A and B:", intersection_set)
```

Intersection of A and B: {3, 4}

```
In [58]: difference_set = set_a - set_b
print("Difference(A - B)", difference_set)
```

Difference(A - B) {1, 2}

```
In [60]: sym_diff_set = set_a ^ set_b
print("Symmetric Difference of A and B", sym_diff_set)
```

Symmetric Difference of A and B {1, 2, 5, 6}

```
In [62]: subset_result = {1,2}.issubset(set_a)
print("Is {1,2} a subset of A?",subset_result)
```

Is {1,2} a subset of A? True

```
In [68]: superset_result = set_a.issuperset({1,2})
print("Is A a superset of {1,2}?",superset_result)
```

Is A a superset of {1,2}? True

```
In [70]: disjoint_result = set_a.isdisjoint({7,8})
print("Are A and {7,8}disjoint?",disjoint_result)
```

Are A and {7,8}disjoint? True

## Set Applications

```
In [73]: list_with_duplicates = [1,2,2,3,4,4,5]
unique_list = list(set(list_with_duplicates))
print("List without duplicates:",unique_list)
```

List without duplicates: [1, 2, 3, 4, 5]

```
In [75]: original_list = [10,20,30,40]
converted_set = set(original_list)
converted_back_list = list(converted_set)
print("List->set -> List:",converted_back_list)
```

List->set -> List: [40, 10, 20, 30]

```
In [77]: list1 = [1,2,3,4]
list2 = [3,4,5,6]
common_elements = list(set(list1) & set(list2))
print("common elements in both lists",common_elements)
```

common elements in both lists [3, 4]

```
In [81]: unique_to_list1 = list(set(list1)-set(list2))

print("Elements in list1 but not in list2:",unique_to_list1)
```

Elements in list1 but not in list2: [1, 2]

```
In [83]: squares_set = {x**2 for x in range(1,6)}
print("Set of sqaures from 1 to 5",squares_set)
```

Set of sqaures from 1 to 5 {1, 4, 9, 16, 25}

## Frozen set program

```
In [ ]: my_frozenset = frozenset([1,2,3,4,5])  
        print("Frozen",my_frozenset)
```