## Set Programming Example:

**Example-1: Write a program to remove the given number from the set?**

```
my_set = {1,2,3,4,5,6,12,24}
n = int(input("Enter the number you want to remove"))
my_set.discard(n)
print("After Removing:",my_set)
```

Output: Enter the number you want to remove:12
After Removing: {1, 2, 3, 4, 5, 6, 24}

**Example-2: Write a program to add multiple elements to the set?**

```
set1 = set([1,2,4,"John","CS"])
set1.update(["Apple","Mango","Grapes"])
print(set1)
```

Output: {1, 2, 4, 'Apple', 'John', 'CS', 'Mango', 'Grapes'}

**Example-3: Write a program to find the union between two set?**

```
>>> set1 = set(["Peter","Joseph", 65,59,96])
>>> set2  = set(["Peter",1,2,"Joseph"])
>>> set3 = set1.union(set2)
>>> print(set3)
```

Output: {96, 65, 2, 'Joseph', 1, 'Peter', 59}

**Example-4: Write a program to find the intersection between two sets?**

```
>>> set1 = {23,44,56,67,90,45,"Javatpoint"}
>>> set2 = {13,23,56,76,"Sachin"}
>>> set3 = set1.intersection(set2)
>>> print(set3)
```

Output: {56, 23}

**Example-5: Write the program to add element to the frozenset?**

```
>>> frozenset_object = frozenset([10,20,30])
>>> frozenset_object.add(40)
```

**AttributeError**: 'frozenset' object has no attribute 'add'

**Note**: Above code raised an error because frozensets are immutable and can't be changed after creation.

**Example - 6: Write the program to find the issuperset, issubset and superset?**

```
>>> set1 = set(["Peter","James","Camroon","Ricky","Donald"])
>>> set2 = set(["Camroon","Washington","Peter"])
```

```
>>> set3 = set(["Peter"])

>>> issubset  =  set1 >= set2
>>> print(issubset)              # False

>>> issuperset  =  set1 <= set2
>>> print(issuperset)            # False

>>> issubset   =   set3 <= set2
>>> print(issubset)                  # True

>>> issuperset  =  set2 >= set3
>>> print(issuperset)                # True
```

## 1. Does converting an object to a set maintain the object's order?

No. A set is not an ordered data structure, so order is not maintained.

Look what happens when we convert the list `[3,2,1]` to a set. It becomes `{1,2,3}`.

```
a = set([3,2,1])
a
Output:  {1, 2, 3}
```

## 2. Check if a set a subset of another set

This can be done with the `issubset()` method.

```
>>>   a = {4,5}
>>>   b = {1,2,3,4,5}
>>>   a.issubset(b)
True
>>>   b.issubset(a)
 False
```

## 3. Check if a set is a subset, using comparison operators

A set, `s1`, is a subset of `s2` if all elements of `s1` are in `s2`.

The operators `<=` will return `True` if all elements of the 1st set exist in the 2nd set (aka. is a subset).

```
>>>   a = {'a','b'}
>>>   b = {'a','b','c'}
```

```
>>>  a <= b
True
>>>  b  <=  a
False
```

**4. Is a set a subset of itself?**

Yes. Because a set contains all elements in itself, it is indeed a subset of itself. This is important to understand when we contrast “subset” with “proper subset” later.

```
>>>  a = {10,20}a.issubset(a)
True
```

**5. Check if a specific value exists in a set**

Like other types of iterables, we can check if a value exists in a set with the `in` operator.

```
>>>  s = {5,7,9}5 in s
True
>>>  6 in s
False
```

**6. Write a Python program to create an empty set.**

```
>>> my_set = set()
```

**7. Write a Python program to create a set with elements "apple", "banana", and "cherry".**

```
>>> fruits = {"apple", "banana", "cherry"}
```

**8. Write a Python program to find the length of a set.**

```
>>> my_set = {1, 2, 3, 4, 5}
>>> len(my_set)
5
```

**9. Write a Python program to add an element to a set.**

```
>>> my_set = {1, 2, 3, 4, 5}
>>> my_set.add(6)
>>> my_set
{1,2,3,6,4,5}
```

**10. Write a Python program to remove an element from a set.**

```
>>> my_set = {1, 2, 3, 6,4, 5}
>>> my_set.remove(3)
```

```
>>> my_set
{1,2,6,4,5}
```

**11. Write a Python program to check if an element is present in a set.**

```
>>> my_set = {1,2,6,4,5}
>>> if 4 in my_set:
        print("Element found")
>>> else:
        print("Element not found")
```

**Output:** Element found

**12. Write a Python program to perform set union.**

```
>>> set1 = {1, 2, 3}
>>> set2 = {3, 4, 5}
>>> union_set = set1.union(set2)
{1,2,3,4,5}
```

**13. Write a Python program to perform set intersection.**

```
>>> intersection_set = set1.intersection(set2)
>>> intersection_set
{3}
```

**14. Write a Python program to perform set difference.**

```
>>> difference_set  =  set1.difference(set2)
>>> difference_set
{1, 2}
```

**15. Write a Python program to perform symmetric difference.**

```
>>> symmetric_difference_set  =  set1.symmetric_difference(set2)
>>> symmetric_difference_set
{1, 2, 4, 5}
```

**16. Write a Python program to check if a set is a subset of another set.**

```
if set1.issubset(set2):
   print("set1 is a subset of set2")
else:
   print("set1 is not a subset of set2")
```

**output**: set1 is not a subset of set2

**17. Write a Python program to check if two sets are disjoint.**

```
if set1.isdisjoint(set2):
   print("Sets are disjoint")
else:
   print("Sets are not disjoint")
```

**Output:** Sets are not disjoint

**18. Write a Python program to clear all elements from a set.**

```
>>> my_set = {1,2,6,4,5}
>>> my_set.clear()
>>> my_set
set()
```

**19. Write a Python program to copy a set.**

```
>>> my_set = {1,2,6,4,5}
>>> copy_set = my_set.copy()
>>> copy_set
{1,2,6,4,5}
```

**20. Write a Python program to remove and return an arbitrary element from a set.**

```
>>> my_set = {1,2,6,4,5}
>>> my_set.pop()
1
```

**21. Write a Python program to find the maximum and minimum elements in a set.**

```
>>> my_set = {2,6,4,5}
>>> max(my_set)
6
>>> min(my_set)
2
```

**22. Write a Python program to find the difference between two sets using the '-' operator.**

```
>>> set1 = {1, 2, 3}
>>> set2 = {3, 4, 5}
>>> difference_set = set1 - set2
>>> difference_set
{1, 2}
```

**23. Write a Python program to update a set with elements from another iterable.**

```
>>> my_set = {2, 4, 5, 6}
>>> new_elements = [6, 7, 8]
```

```
>>> my_set.update(new_elements)
>>> my_set
{2, 4, 5, 6, 7, 8}
```

**24. Write a Python program to remove a specific element from a set using the discard method.**

```
>>> my_set = {2, 4, 5, 6, 7, 8}
>>> my_set.discard(6)
>>> my_set
{2,4,5,7,8}
```

**25. Write a Python program to find the common elements between multiple sets.**

```
>>> set1 = {1, 2, 3}
>>> set2 = {3, 4, 5}
>>> common_elements  =  set.intersection(set1, set2)
>>> common_elements
{3}
```

**26. Write a Python program to remove the intersection of two sets from one set.**

```
>>> set1 = {1, 2, 3}
>>> set2 = {3, 4, 5}
>>> set1.difference_update(set2)
{1, 2}
```

**27. Write a Python program to create a frozen set.**

```
>>> frozen_set = frozenset([1, 2, 3])
>>> frozen_set
frozenset({1, 2, 3})
```

**28. Write a Python program to convert a list into a set.**

```
>>> my_list = [1, 2, 3, 4, 5]
>>> my_set = set(my_list)
>>> my_set
{1, 2, 3, 4, 5}
```

**29. Write a Python program to find the union of multiple sets.**

```
>>> union_set = set.union(set1, set2, set3)
```

**30. Write a Python program to check if two sets are equal.**

```
>>> set1 = {1,2,3}
>>> set2 = {3,4,5}
>>> if set1 == set2:
```

```
        print("Sets are equal")
>>> else:
        print("Sets are not equal")
```
**Output**: Sets are not equal