

Semester	S.E. Semester IV – Information Technology
Subject	Python Lab
Subject Professor In-charge	Mr. Ichhanshu Jaiswal
Assisting Teachers	Mr. Ichhanshu Jaiswal
Laboratory	

Class Name	SE INFT B	
Student Name	Yash Gupta	
Grade and Subject Teacher's Signature		

Experiment Number	5	
Experiment Title	Implement Set Methods and Frozenset Methods.	
Resources / Apparatus Required	Hardware:	Software:
Objectives (Skill Set / Knowledge Tested / Imparted)		
Theory and Code	<p>Sets and frozensets are both built-in data types in Python for storing collections of unique elements. Sets are mutable, meaning they can be modified after creation, while frozensets are immutable, meaning they cannot be modified after creation.</p> <p>Set methods include:</p> <p>add(): adds an element to the set</p> <p>remove(): removes an element from the set</p> <p>intersection(): returns the common elements between two sets</p> <p>union(): returns the combined set of two or more sets</p> <p>difference(): returns the difference between two sets</p>	

`symmetric_difference()`: returns the symmetric difference between two sets

`pop()`: removes and returns an arbitrary element from the set

`clear()`: removes all elements from the set

Frozenset methods include:

`intersection()`: returns the common elements between two frozensets

`union()`: returns the combined frozenset of two or more frozensets

`difference()`: returns the difference between two frozensets

`symmetric_difference()`: returns the symmetric difference between two frozensets

`copy()`: returns a shallow copy of the frozenset

`frozenset()`: returns a new frozenset with the specified iterable as its elements

`isdisjoint()`: returns True if two frozensets have no common elements, otherwise False

`issubset()`: returns True if all elements of a frozenset are in another frozenset, otherwise False

`issuperset()`: returns True if all elements of another frozenset are in a frozenset, otherwise False

These methods can be used to perform a variety of operations on sets and frozensets, such as finding common elements, finding the difference between sets, and modifying sets.

Code:

```
# Create a set
```

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

```
# Demonstrate set methods

print("Original set:", my_set)

# add() method

my_set.add(6)

print("After add():", my_set)

# remove() method

my_set.remove(3)

print("After remove():", my_set)

# intersection() method

new_set = {4, 5, 6, 7, 8}

intersection_set = my_set.intersection(new_set)

print("Intersection of sets:", intersection_set)

# pop() method

popped_item = my_set.pop()

print("Popped item:", popped_item)

print("After pop():", my_set)

# Create a frozenset

my_frozenset = frozenset({1, 2, 3, 4, 5})

# Demonstrate frozenset methods

print("Original frozenset:", my_frozenset)

# intersection() method

intersection_frozenset = my_frozenset.intersection({4, 5, 6, 7, 8})
```

```
print("Intersection of frozensets:", intersection_frozenset)

# difference() method

difference_frozenset = my_frozenset.difference({2, 4, 6})

print("Difference of frozensets:", difference_frozenset)

# len() function

length = len(my_frozenset)

print("Length of frozenset:", length)
```

Output

```
PS C:\Users\yashg\OneDrive\Desktop\Python> & C:/Users/yashg/AppData/Local/Programs/Python/
codexx.py
Original set: {1, 2, 3, 4, 5}
After add(): {1, 2, 3, 4, 5, 6}
After remove(): {1, 2, 4, 5, 6}
Intersection of sets: {4, 5, 6}
Popped item: 1
After pop(): {2, 4, 5, 6}
Original frozenset: frozenset({1, 2, 3, 4, 5})
Intersection of frozensets: frozenset({4, 5})
Difference of frozensets: frozenset({1, 3, 5})
Length of frozenset: 5
PS C:\Users\yashg\OneDrive\Desktop\Python>
```

Conclusion

The Python program creates and manipulates a set and a frozenset, demonstrating various methods available for each data type. The set is mutable and has methods like `add()` and `remove()`, while the frozenset is immutable and has methods like `intersection()` and `difference()`. The program showcases how sets and frozensets can be useful for storing unique elements and

performing operations on them.