**Q1. What is a set in Python?**

**Ans:** A set in Python is an unordered collection of unique elements. Sets are mutable, meaning you can add or remove elements, but they do not allow duplicate values. They are useful for operations that require uniqueness, such as removing duplicates from a list or performing mathematical set operations (like unions and intersections).

**Q2.How do you create a set in Python?**

**Ans:** You can create a set in Python using two main methods: using curly braces {} or using the set() constructor.

**Using curly braces**

my\_set = {1, 2, 3, 4, 5}

print(my\_set)

# Output: {1, 2, 3, 4, 5}

**Using set() constructor**

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

my\_set = set(my\_list)

print(my\_set)

# Output: {1, 2, 3, 4}

**Q3. How do you add elements to a set in Python?**

**Ans:** We can add elements to a set in Python using the add() method or the update() method**.**

**Using add() method**

The add() method adds a single element to the set. If the element already exists in the set, it will not be added again (no duplicates).

my\_set = {1, 2, 3}

my\_set.add(4)

print(my\_set)

# Output: {1, 2, 3, 4}

my\_set.add(2)

print(my\_set)

# Output: {1, 2, 3, 4} (unchanged)

### **Using update()**

The update() method can add multiple elements to the set at once. You can pass an iterable (like a list, tuple, or another set) to this method.

my\_set = {1, 2, 3}

my\_set.update([4, 5])

print(my\_set)

# Output: {1, 2, 3, 4, 5}

my\_set.update({6, 7})

print(my\_set)

# Output: {1, 2, 3, 4, 5, 6, 7}

**Q4. How do you remove elements from a set in Python?**

We can remove elements from a set in Python using several methods: remove(), discard(), and pop().

**Using remove() method**

The remove() method removes a specified element from the set. If the element is not found, it raises a KeyError.

my\_set = {1, 2, 3, 4}

my\_set.remove(3)

print(my\_set)

# Output: {1, 2, 4}

**Using discard() method**

The discard() method also removes a specified element from the set, but it does not raise an error if the element is not found.

my\_set = {1, 2, 3, 4}

my\_set.discard(3)

print(my\_set)

# Output: {1, 2, 4}

my\_set.discard(5)

print(my\_set)

# Output: {1, 2, 4}

**Using pop() method**

The pop() method removes and returns an arbitrary element from the set. If the set is empty, it raises a KeyError.

my\_set = {1, 2, 3, 4}

removed\_element = my\_set.pop()

print(removed\_element)

print(my\_set)

# Output: Remaining elements (e.g., {2, 3, 4})

**Q5. How do you get the length of a set in Python?**

We can get the length of a set in Python using the built-in len() function. This function returns the number of unique elements in the set.

my\_set = {1, 2, 3, 4, 5}

length = len(my\_set)

print(length)

# Output: 5