HashSet in Java implements the Set interface of the Collections Framework. It is used to store the unique elements, and it doesn't maintain any specific order of elements.

- HashSet does not allow duplicate elements.
- Uses HashMap internally which is an implementation of hash table data structure.
- Also implements Serializable and Cloneable interfaces.
- HashSet is not thread-safe. To make it thread-safe, synchronization is needed externally.
- Does not support primitive types directly; requires wrapper classes (Integer, Character, etc.).

```
import java.util.*;


class GFG
{
   public static void main(String[] args)
   {
      // Instantiate an object of HashSet
      HashSet<Integer> hs = new HashSet<>();


           // Adding elements
      hs.add(1);
      hs.add(2);
      hs.add(1);


      System.out.println("HashSet Size: " + hs.size());
      System.out.println("Elements in HashSet: " + hs);
   }
```

```
}
```

Constructors of HashSet class

To create a HashSet, we need to create an object of the HashSet class. The HashSet class consists of various constructors that allow the possible creation of the HashSet. The following are the constructors available in this class.

1. HashSet()

Creates a new empty HashSet with default capacity (16) and load factor (0.75).

HashSet<String> set = new HashSet<>();

2. HashSet(int initialCapacity)

Creates an empty HashSet with the specified initial capacity and default load factor (0.75).

HashSet<Type> set = new HashSet<>(initialCapacity);

3. HashSet(int initialCapacity, float loadFactor)

Creates an empty HashSet with the given initial capacity and load factor.

HashSet<Type> set = new HashSet<>(initialCapacity, loadFactor);

4. HashSet(Collection<? extends E> c)

Creates a new HashSet containing the elements of the specified collection (removes duplicates automatically).

HashSet<Type> set = new HashSet<>(c);

Performing Various Operations on HashSet

Let's see how to perform a few frequently used operations on the HashSet.

1. Adding Elements in HashSet

To add an element to the HashSet, we can use the add() method. However, the insertion order is not retained in the HashSet. We need to

keep a note that duplicate elements are not allowed and all duplicate elements are ignored.