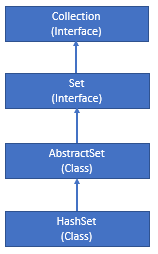
**Java HashSet**

HashSet class implements Set interface. HashSet allows only unique values with random insertion order.



**Notes:**

1. HashSet is not synchronized.
2. HashSet doesn’t allow to store duplicate values doing so will override the last value.
3. HashSet elements doesn’t maintain any order and return random order.
4. HashSet allows only single null value to store and it will override existing null value on adding further null values.
5. HashSet doesn’t allow duplicate values, it will override existing value on adding further same values.

**Constructors:**

Below are the constructors help us to create HashSet object:

**HashSet() →** Default initial capacity is 16 and default load factor is 0.75.

HashSet hashSet = new HashSet();

**HashSet(Collection c)** → Create HashSet collection using specified collection.

HashSet hashSet = new HashSet(Collection C);

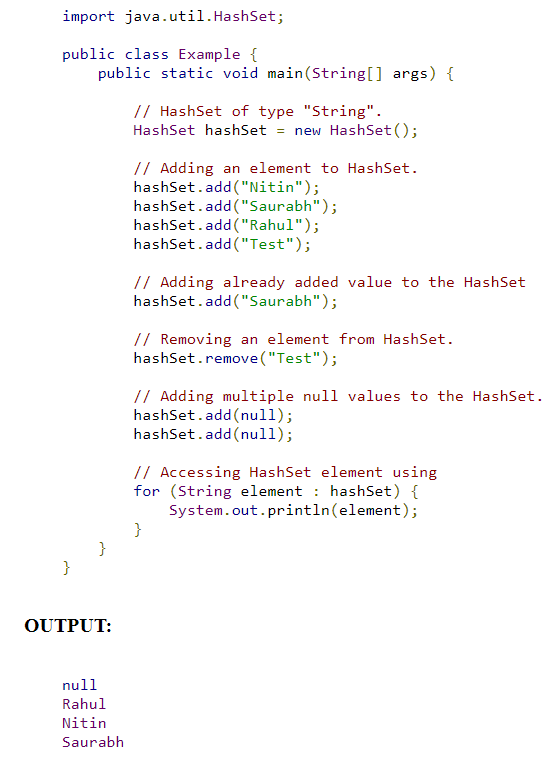
**HashSet(int initialCapacity) →** Create HashSet with specified intial capacity and default 0.75 load factor.

HashSet hashSet = new HashSet(int initialCapacity);

**HashSet(int initialCapacity, float loadFactor) →**Create HashSet with specified with specified initial capacity and load factor.

HashSet hashSet = new HashSet(int initialCapacity, float loadFactor);

**Example:**



**HashSet Methods:**

Below are some of the methods from HashSet class.

**add():**Method helps us to add an element in HashSet.

**remove():**Method helps us to remove an element from HashSet.

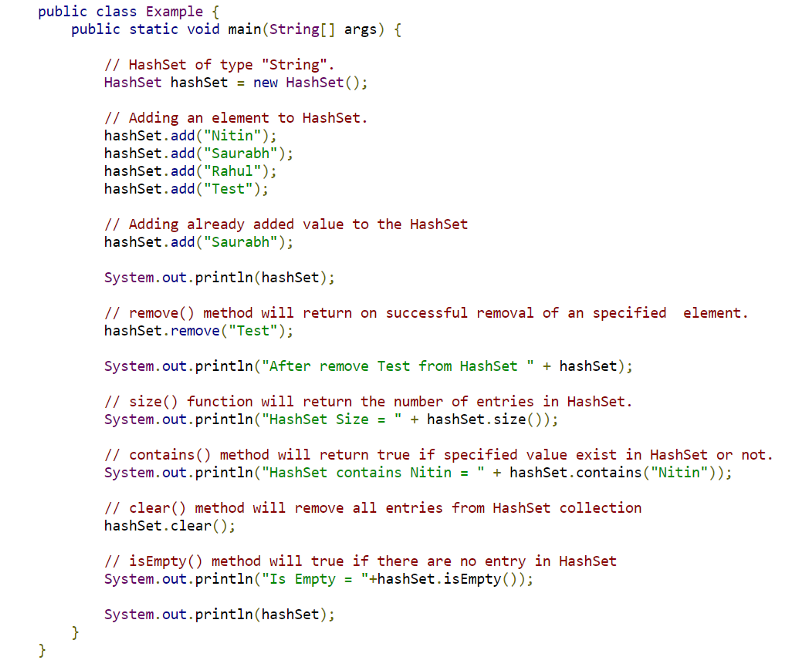
**size():**Method returns the number of element in HashSet Collection.

**remove():**Method helps us to remove an element from HashSet.

**contains():**Method returns true if specified element present in HashSet collection else return false.

**clear():**Method helps us to remove all the elements from HashSet.

**isEmpty():**Method returns true if number of elements is zero else return false.



**OUTPUT:**

