

Java HashSet class implements the Set interface, backed by a hash table which is actually a [HashMap](#) instance.

No guarantee is made as to the iteration order of the hash sets which means that the class does not guarantee the constant order of elements over time.

This class permits the null element.

The class also offers constant time performance for the basic operations like add, remove, contains, and size assuming the hash function disperses the elements properly among the buckets.

Java HashSet Features

A few important features of HashSet are mentioned below:

- Implements [Set Interface](#).
- The underlying data structure for HashSet is [Hashtable](#).
- As it implements the Set Interface, duplicate values are not allowed.
- Objects that you insert in HashSet are not guaranteed to be inserted in the same order. Objects are inserted based on their hash code.
- NULL elements are allowed in HashSet.
- HashSet also implements **Serializable** and **Cloneable** interfaces.

```
public class HashSet<E> extends AbstractSet<E> implements Set<E>, Cloneable, Serializable
Sample Input and Output:
5
90
56
45
78
25
78
Sample Output:
78 was found in the set.
Sample Input and output:
3
2
7
9
5
Sample Input and output:
5 was not found in the set.
```

Answer: (penalty regime: 0 %)

Reset answer

```
1 import java.util.HashSet;
2 import java.util.Scanner;
3
4 class prog {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7         int n = sc.nextInt();
8
9         // Create a HashSet object called numbers
10        HashSet<Integer> numbers = new HashSet<>();
11
12        // Add values to the set
13        for (int i = 0; i < n; i++) {
14            numbers.add(sc.nextInt());
15        }
16
17        int skey = sc.nextInt();
18
19        // Check if skey is in the set and show the result
20        if (numbers.contains(skey)) {
21            System.out.println(skey + " was found in the set.");
22        } else {
23            System.out.println(skey + " was not found in the set.");
24        }
25
26        // Display which numbers between 1 and 10 are in the set
27        for (int i = 1; i <= 10; i++) {
28            if (numbers.contains(i)) {
29            }
30        }
31
32        sc.close();
33    }
34 }
```

	Test	Input	Expected	Got	
✓	1	5 90 56 45 78 25 78	78 was found in the set.	78 was found in the set.	✓
✓	2	3 -1 2 4 5	5 was not found in the set.	5 was not found in the set.	✓

Passed all tests! ✓

Write a Java program to compare two sets and retain elements that are the same.

Sample Input and Output:

5

Football

Hockey

Cricket

Volleyball

Basketball

7 // **HashSet 2:**

Golf

Cricket

Badminton

Football

Hockey

Volleyball

Handball

SAMPLE OUTPUT:

Football

Hockey

Cricket

Volleyball

Basketball

Answer: (penalty regime: 0 %)

```
1 import java.util.HashSet;
2 import java.util.Scanner;
3
4 public class Main {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         // Read the first set
9         int n1 = sc.nextInt();
10        sc.nextLine(); // Consume newline character
11        HashSet<String> set1 = new HashSet<>();
12        for (int i = 0; i < n1; i++) {
13            set1.add(sc.nextLine());
14        }
15
16        // Read the second set
17        int n2 = sc.nextInt();
18        sc.nextLine(); // Consume newline character
19        HashSet<String> set2 = new HashSet<>();
20        for (int i = 0; i < n2; i++) {
21            set2.add(sc.nextLine());
22        }
23
24        // Retain only elements that are present in both sets
25        set1.retainAll(set2);
26
27        // Output the result
28        for (String element : set1) {
29            System.out.println(element);
30        }
31
32        sc.close();
33    }
34 }
```

	Test	Input	Expected	Got	
✓	1	5 Football Hockey Cricket Volleyball Basketball 7 Golf Cricket Badminton Football Hockey Volleyball Throwball	Cricket Hockey Volleyball Football	Cricket Hockey Volleyball Football	✓
✓	2	4 Toy Bus Car Auto 3 Car Bus Lorry	Bus Car	Bus Car	✓

Passed all tests! ✓

Java HashMap Methods

[containsKey\(\)](#). Indicate if an entry with the specified key exists in the map[containsValue\(\)](#). Indicate if an entry with the specified value exists in the map[putIfAbsent\(\)](#). Write an entry into the map but only if an entry with the same key does not already exist[remove\(\)](#). Remove an entry from the map[replace\(\)](#) Write to an entry in the map only if it exists[size\(\)](#). Return the number of entries in the map

Your task is to fill the incomplete code to get desired output

Answer: (penalty regime: 0 %)

Reset answer

```

1 import java.util.HashMap;
2 import java.util.Map.Entry;
3 import java.util.Set;
4 import java.util.Scanner;
5
6 class prog {
7     public static void main(String[] args) {
8         // Creating HashMap with default initial capacity and load factor
9         HashMap<String, Integer> map = new HashMap<String, Integer>();
10
11         String name;
12         int num;
13         Scanner sc = new Scanner(System.in);
14         int n = sc.nextInt();
15         for (int i = 0; i < n; i++) {
16             name = sc.next();
17             num = sc.nextInt();
18             map.put(name, num);
19         }
20
21         // Printing key-value pairs
22         Set<Entry<String, Integer>> entrySet = map.entrySet();
23         for (Entry<String, Integer> entry : entrySet) {
24             System.out.println(entry.getKey() + " : " + entry.getValue());
25         }
26
27         System.out.println("-----");
28
29         // Creating another HashMap
30         HashMap<String, Integer> anotherMap = new HashMap<String, Integer>();
31
32         // Inserting key-value pairs to anotherMap using put() method
33         anotherMap.put("SIX", 6);
34         anotherMap.put("SEVEN", 7);
35
36         // Inserting key-value pairs of map to anotherMap using putAll() method
37         anotherMap.putAll(map); // Fill in this line
38
39         // Printing key-value pairs of anotherMap
40         entrySet = anotherMap.entrySet();
41         for (Entry<String, Integer> entry : entrySet) {
42             System.out.println(entry.getKey() + " : " + entry.getValue());
43         }
44
45         // Adds key-value pair 'FIVE-5' only if it is not present in map
46         map.putIfAbsent("FIVE", 5);
47
48         // Retrieving a value associated with key 'TWO'
49         int value = map.getDefault("TWO", -1); // Use getDefault in case "TWO" is not in map
50         System.out.println(value);
51
52         // Checking whether key 'ONE' exists in map

```

	Test	Input	Expected	Got	
✓	1	3 ONE 1 TWO 2 THREE 3	ONE : 1 TWO : 2 THREE : 3 ----- SIX : 6 ONE : 1 TWO : 2 SEVEN : 7 THREE : 3 2 true true 4	ONE : 1 TWO : 2 THREE : 3 ----- SIX : 6 ONE : 1 TWO : 2 SEVEN : 7 THREE : 3 2 true true 4	✓

Passed all tests! ✓

◀ Lab-11-MCQ

Jump to...



TreeSet example ▶