

Question 1

Correct

Marked out of 1.00

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
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 sc.close();
27 }
28 }

```

	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! ✓

Question 2

Correct

Marked out of 1.00

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 CompareSets {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         // Input first set
9
10        int n1 = sc.nextInt();
11        sc.nextLine(); // Consume newline
12        HashSet<String> set1 = new HashSet<>();
13
14        for (int i = 0; i < n1; i++) {
15            set1.add(sc.nextLine());
16        }
17
18        // Input second set
19
20        int n2 = sc.nextInt();
21        sc.nextLine(); // Consume newline
22        HashSet<String> set2 = new HashSet<>();
23
24        for (int i = 0; i < n2; i++) {
25            set2.add(sc.nextLine());
26        }
27    }
28 }
```

```
28 // Retain only common elements in set1
29 set1.retainAll(set2);
30
31 // Output common elements
32
33 for (String sport : set1) {
34     System.out.println(sport);
35 }
36
37 sc.close();
38 }
39 }
40
41
```

	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! ✓

Question 3

Correct

Marked out of 1.00

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

```
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     System.out.println("-----");
27
28     // Creating another HashMap
29     HashMap<String, Integer> anotherMap = new HashMap<String, Integer>();
30
31     // Inserting key-value pairs to anotherMap using put() method
32     anotherMap.put("SIX", 6);
33     anotherMap.put("SEVEN", 7);
34
35     // Inserting key-value pairs of map to anotherMap using putAll() method
36     anotherMap.putAll(map); // Code to add all entries from 'map' to 'anotherMap'
37
38     // Printing key-value pairs of anotherMap
39     entrySet = anotherMap.entrySet();
40     for (Entry<String, Integer> entry : entrySet) {
41         System.out.println(entry.getKey() + " : " + entry.getValue());
42     }
43
44     // Adds key-value pair 'FIVE-5' only if it is not present in map
45     map.putIfAbsent("FIVE", 5);
46
47     // Retrieving a value associated with key 'TWO'
48     int value = map.getOrDefault("TWO", -1); // Using getOrDefault to avoid null if 'TWO'
49     System.out.println(value);
50
51     // Checking whether key 'ONE' exists in map
52     System.out.println(map.containsKey("ONE"));
53
54     // Checking whether value '3' exists in map
55     System.out.println(map.containsValue(3));
56
57     // Retrieving the number of key-value pairs present in map
58     System.out.println(map.size());
59
60     sc.close();
61 }
62 }
63 }
```

© 2011 Pearson Education, Inc. or its affiliate(s). All rights reserved. This material is intended solely for the personal use of the individual user and is not to be disseminated broadly.

Passed all tests! ✓

Jump to...

TreeSet example ►