

## Week-11

### Set, Map

231901029  
Madhesh M A

#### Program 1:

```
import java.util.HashSet;
import java.util.Set;
import java.util.Scanner;
class prog {
    public static void main(String[] args){
        Scanner sc= new Scanner(System.in);
        int n = sc.nextInt();
        // Create a HashSet object called numbers
        Set<Integer> numbers = new HashSet<>();

        // Add values to the set
        for(int i=0;i<n;i++)
            numbers.add(sc.nextInt());

        int skey=sc.nextInt();

        // Show which numbers between 1 and 10 are in the set

        if(numbers.contains(skey)){
            System.out.println(skey + " was found in the set.");
        }
        else{
            System.out.println(skey + " was not found in the set.");
        }
    }
}
```

---

## Program 2:

```
import java.util.*;

public class prog {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Read the size of the first set
        int a = sc.nextInt();
        sc.nextLine(); // Consume the leftover newline
        Set<String> set1 = new HashSet<>();

        // Read elements for the first set
        for (int i = 0; i < a; i++) {
            set1.add(sc.nextLine());
        }

        // Read the size of the second set
        int b = sc.nextInt();
        sc.nextLine(); // Consume the leftover newline
        Set<String> set2 = new HashSet<>();

        // Read elements for the second set
        for (int i = 0; i < b; i++) {
            set2.add(sc.nextLine());
        }

        // Retain only the common elements between the two sets
        Set<String> retained = new HashSet<>(set1);
        retained.retainAll(set2);

        // Print the retained elements
        for (String sport : retained) {
            System.out.println(sport);
        }
    }
}
```

```
}  
}
```

---

### Program 3:

```
import java.util.HashMap;  
import java.util.Map.Entry;  
import java.util.Set;  
import java.util.Scanner;  
  
class prog {  
    public static void main(String[] args) {  
        // Creating HashMap with default initial capacity and load factor  
        HashMap<String, Integer> map = new HashMap<>();  
  
        String name;  
        int num;  
        Scanner sc = new Scanner(System.in);  
        int n = sc.nextInt();  
  
        // Reading key-value pairs from user input  
        for (int i = 0; i < n; i++) {  
            name = sc.next();  
            num = sc.nextInt();  
            map.put(name, num);  
        }  
  
        // Printing key-value pairs  
        Set<Entry<String, Integer>> entrySet = map.entrySet();  
        for (Entry<String, Integer> entry : entrySet) {  
            System.out.println(entry.getKey() + " : " + entry.getValue());  
        }  
  
        System.out.println("-----");  
  
        // Creating another HashMap  
        HashMap<String, Integer> anotherMap = new HashMap<>();
```

```

// Inserting key-value pairs to anotherMap using put() method
anotherMap.put("SIX", 6);
anotherMap.put("SEVEN", 7);

// Inserting key-value pairs of map to anotherMap using putAll() method
anotherMap.putAll(map); // Corrected line

// Printing key-value pairs of anotherMap
entrySet = anotherMap.entrySet();
for (Entry<String, Integer> entry : entrySet) {
    System.out.println(entry.getKey() + " : " + entry.getValue());
}

// Adds key-value pair 'FIVE-5' only if it is not present in map
map.putIfAbsent("FIVE", 5);

// Retrieving a value associated with key 'TWO'
Integer value = map.get("TWO"); // Changed to Integer to handle null
if (value != null) {
    System.out.println(value);
} else {
    System.out.println("Key 'TWO' not found");
}

// Checking whether key 'ONE' exists in map
System.out.println(map.containsKey("ONE"));

// Checking whether value '3' exists in map
System.out.println(map.containsValue(3));

// Retrieving the number of key-value pairs present in map
System.out.println(map.size());
}
}

```