### **Set Operations Code Explanation**

## **Overview**

This code implements three common set operations: **Union, Intersection, and Complement**..

## **Classes in the Code**

### **SetOperations Class**

This class contains methods to compute:

* **Union**: Combines elements from two sets without duplicates.
* **Intersection**: Finds common elements between two sets.
* **Complement**: Finds elements in one set that are not in another.

### **Methods in SetOperations**

#### **getUnion(int[] userInput1, int[] userInput2)**

**Purpose**: Computes the union of two arrays, ensuring unique elements.

* **Parameters**:
  + userInput1: First array of integers.
  + userInput2: Second array of integers.
* **Process**:
  + Adds all elements from userInput1.
  + Iterates through userInput2, adding only non-duplicate elements.
* **Returns**: A list containing the union of the two arrays.

#### **getIntersection(int[] userInput1, int[] userInput2)**

**Purpose**: Computes the intersection of two arrays.

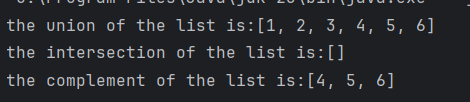
* **Parameters**:
  + userInput1: First array of integers.
  + userInput2: Second array of integers.
* **Process**:
  + Loops through userInput1.
  + Checks if each element exists in userInput2 and is not already added.
* **Returns**: A list containing the common elements between the two arrays.

#### **getComplement(int[] userInput1, int[] userInput2)**

**Purpose**: Computes the complement of userInput1 relative to userInput2.

* **Parameters**:
  + userInput1: Universal set.
  + userInput2: Subset whose elements should be removed from userInput1.
* **Process**:
  + Loops through userInput1 and adds elements that are not in userInput2.
* **Returns**: A list containing elements from userInput1 that are not in userInput2.

#### **Results:**



## **Conclusion**

This program helps understanding set operations easier. Helps understanding basic iterations and list manipulation in java easier.