1. Create a method that takes an ArrayList of String and returns a copy of that ArrayList with no duplicates.

The relative ordering of elements in the new ArrayList should be the same.

Sample Input: {"qwerty", "asdfgh", "qwer", "123", "qwerty", "123", "zxcvbn", "asdfgh"}

Sample Output: {"qwerty", "asdfgh", "qwer", "123", "zxcvbn"}

**Answer**:

import java.util.\*;  
public class Main  
{  
 public static void main(String[] args) {  
   
ArrayList<String>  
list = new ArrayList<>(  
Arrays  
.asList("qwerty", "asdfgh", "qwer", "123", "qwerty", "123", "zxcvbn", "asdfgh"));  
   
System.out.println("ArrayList with duplicates: "  
+ list);  
   
ArrayList<String>  
newList = removeDuplicates(list);  
   
System.out.println("ArrayList with duplicates removed: "  
+ newList);  
}  
   
   
public static <T> ArrayList<T> removeDuplicates(ArrayList<T> list){  
ArrayList<T> newList = new ArrayList<T>();  
for (T element : list) {  
if (!newList.contains(element)){   
newList.add(element);  
}  
}  
return newList;  
}  
   
}

2. Use a HashSet to do the following:

Create a program that takes as user input an integer n.

Then prompts n strings from the user.

Afterward, print all the Strings without duplicates in any order (print at most 1 of each String).

import java.util.HashSet;  
import java.util.Scanner;  
  
public class main{  
  
 public static void main(String[] args) {  
 HashSet<String > set=new HashSet<String>();  
 Scanner sc=new Scanner(System.in);  
 System.out.print("Enter the value of n: ");  
 int n=sc.nextInt();  
 System.out.println("Enter n strings: ");  
 for(int i=0;i<n;i++){  
 String str=sc.next();  
 set.add(str);  
 }   
 for(String str:set){  
 System.out.println(str);  
 }  
 }  
}

3. Determine the output: **14**

**4.** a. Create a class named RectangularPrism that has instance variables length, width, and height.

b. Create a constructor that sets those variables. Also, create get and set methods for each variable.

c. Create a toString() method.

d. Create an equals() method.

e. Create a hashCode() method.

f. Create a HashSet of RectangularPrism and insert a few elements including duplicates.

Verify that the HashSet removes all the duplicates.

**Answer:** public class RectangularPrism {  
 private int length;  
 private int width;  
 private int height;  
  
 public RectangularPrism(int length, int width, int height) {  
 this.length = length;  
 this.width = width;  
 this.height = height;  
 }  
  
 public int getLength() {  
 return length;  
 }  
  
 public void setLength(int length) {  
 this.length = length;  
 }  
 public int getWidth() {  
 return width;  
 }  
 public void setWidth(int width) {  
 this.width = width;  
 }  
 public int getHeight() {  
 return height;  
 }  
 public void setHeight(int height) {  
 this.height = height;  
 }  
 @Override  
 public String toString() {  
 return "RectangularPrism [length=" + length + ", width=" + width + ", height=" + height + "]";  
 }  
 @Override  
 public int hashCode() {  
 final int prime = 31;  
 int result = 1;  
 result = prime \* result + height;  
 result = prime \* result + length;  
 result = prime \* result + width;  
 return result;  
 }  
 @Override  
 public boolean equals(Object obj) {  
 if (this == obj)  
 return true;  
 if (obj == null)  
 return false;  
 if (getClass() != obj.getClass())  
 return false;  
 RectangularPrism other = (RectangularPrism) obj;  
 if (height != other.height)  
 return false;  
 if (length != other.length)  
 return false;  
 if (width != other.width)  
 return false;  
 return true;  
 }  
  
}

import java.util.HashSet;  
  
public class Driver {  
  
 public static void main(String[] args) {  
 HashSet<RectangularPrism> rectangularPrismSet = new HashSet<RectangularPrism>();  
 RectangularPrism rectangularPrism1 = new RectangularPrism(10,20,30);  
 RectangularPrism rectangularPrism2 = new RectangularPrism(40,50,60);  
 RectangularPrism rectangularPrism3 = new RectangularPrism(10,20,30);  
 RectangularPrism rectangularPrism4 = new RectangularPrism(10,20,30);  
 rectangularPrismSet.add(rectangularPrism1);  
 rectangularPrismSet.add(rectangularPrism2);  
 rectangularPrismSet.add(rectangularPrism3);  
 rectangularPrismSet.add(rectangularPrism4);  
 System.out.println(rectangularPrismSet);  
 }  
}

5. Create a toString() method inside the Pair class.

b. Create an equals() method inside the Pair class.

c. Create a hashCode() method inside the Pair class.

d. Create a HashSet of Pair<Integer, String> and insert a few elements including duplicates.

Verify that the HashSet removes all the duplicates.

import java.util.Objects;  
  
public class A {  
 String str;  
 A(String str) {  
 this.str = str;  
 }  
 public String toString(){  
 return str;  
 }  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 A a = (A) o;  
 return Objects.equals(str, a.str);  
 }  
  
 public int hashCode() {  
 return Objects.hash(str);  
 }  
}  
B.java:  
  
import java.util.Objects;  
  
public class B {  
 String str;  
 B(String str) {  
 this.str = str;  
 }  
 public String toString() {  
 return str;  
 }  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 B b = (B) o;  
 return Objects.equals(str, b.str);  
 }  
  
 public int hashCode() {  
 return Objects.hash(str);  
 }  
}  
Pair.java:  
  
import java.util.Objects;  
class Pair  
  
{  
 public A first;  
 public B second;  
 public Pair(A a, B b)  
 {  
 first = a;  
 second = b;  
 }  
 public void setFirst(A a)  
 {  
 first = a;  
 }  
 public A getFirst()  
 {  
 return first;  
 }  
 public void setSecond(B b)  
 {  
 second = b;  
 }  
 public B getSecond()  
 {  
 return second;  
 }  
 public String toString() {  
 return "Pair{" +  
 "first=" + first +  
 ", second=" + second +  
 '}';  
 }  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 Pair pair = (Pair) o;  
 return Objects.equals(first, pair.first) &&  
 Objects.equals(second, pair.second);  
 }  
 public int hashCode() {  
 return Objects.hash(first, second);  
 }  
}  
Driver.java:  
  
import java.util.HashSet;  
  
public class Driver {  
 public static void main(String[] args) {  
 HashSet<Pair> hashSet = new HashSet<>();  
 hashSet.add(new Pair(new A("A0"), new B("B0")));  
 hashSet.add(new Pair(new A("A1"), new B("B1")));  
 hashSet.add(new Pair(new A("A1"), new B("B1")));  
 hashSet.add(new Pair(new A("A2"), new B("B2")));  
 hashSet.add(new Pair(new A("A2"), new B("B2")));  
 for (Pair p: hashSet) {  
 System.out.println(p);  
 }  
 }  
}