Duplicate elements in ArrayList by avoiding Distinct elements

In an Array list I mean to show is the elements which are repeating and by avoiding the distinct elements .To show this we require a piece of code.  
**Example:**  
package Test;  
import java.util.ArrayList;  
import java.util.HashSet;  
import java.util.List;  
import java.util.Set;  
  
public class DuplicateNumbers {  
  
    public static void main(String[] args) {  
      
     ArrayList<String>list = new ArrayList<String>();  
  
     //list of numbers added in from 0-9  
       
        for (int i = 0; i < 10; i++) {  
            list.add(String.valueOf(i));  
        }  
        //another set of numbers from 0-5  
        for (int i = 0; i < 5; i++) {  
            list.add(String.valueOf(i));  
        }  
  
        System.out.println("My List : " + list);  
        System.out.println("\nHere are the duplicate elements from list : " + findDuplicates(list));  
    }  
  
    public static Set<String> findDuplicates(List<String> listContainingDuplicates) {  
  
        final Set<String> setToReturn = new HashSet<String>();  
        final Set<String> set1 = new HashSet<String>();  
  
        for (String yourInt : listContainingDuplicates) {  
            if (!set1.add(yourInt)) {  
                setToReturn.add(yourInt);  
            }  
        }  
        return setToReturn;  
    }  
}  
  
  
  
**Output:**  
  
My List : [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4]  
  
Here are the duplicate elements from list : [3, 2, 1, 0, 4]

### 2. Reverse a number in java

package Test;  
  
public class ReverseNumber {  
   
 public int numberReverse(int number){  
   
 int reverse = 0;  
        while(number != 0){  
            reverse = (reverse\*10)+(number%10);  
            number = number/10;  
        }  
        return reverse;  
 }  
public static void main(String args[]){  
 ReverseNumber nr =new ReverseNumber();  
   
 System.out.println("Reversed number"+nr.numberReverse(155624));  
}  
}

**Output:**

Reversed number426551

### 3. Sorting Example for a Collection or Array of strings

          A list of strings to be sorted by using the array and by using Array list. To describe it a small piece of code is required.  
  
**Example:**  
  
package Test;  
  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.Collections;  
import java.util.List;  
  
public class StringSortExample {  
  
 public static void main(String[] args) throws Exception {  
  
 String[] strArray = { "Abhishek", "selenium", "Automation" };  
  
 //displaying the string array  
  
 displayArray(strArray);  
  
   ///To sort an array  
  
 Arrays.sort(strArray);  
  
 //after sorting displaying array  
  
 displayArray(strArray);  
  
  //sorted array with case insensitive  
    
Arrays.sort(strArray, String.CASE\_INSENSITIVE\_ORDER);  
   
 displayArray(strArray);  
  
 System.out.println("---------------");  
   
        ///sorting strings using arraylist  
   
 List<String> strList = new ArrayList<String>();  
 strList.add("Abhishek");  
 strList.add("Webdriver");  
 strList.add("Automation");  
 displayList(strList);  
        
 Collections.sort(strList);  
 displayList(strList);  
  
 Collections.sort(strList, String.CASE\_INSENSITIVE\_ORDER);  
 displayList(strList);  
 }  
  
 public static void displayArray(String[] array) {  
 for (String str : array) {  
 System.out.print(str + " ");  
 }  
 System.out.println();  
 }  
  
 public static void displayList(List<String> list) {  
 for (String str : list) {  
 System.out.print(str + " ");  
 }  
 System.out.println();  
 }  
  
}  
  
**Output:**  
  
Abhishek selenium Automation  
Abhishek Automation selenium  
Abhishek Automation selenium  
---------------  
Abhishek Webdriver Automation  
Abhishek Automation Webdriver  
Abhishek Automation Webdriver

4.

### How to Reverse a String using String Buffer and Without using string Buffer

package Test;  
  
public class ReverseString {  
 public String reverseMyString(String str){   
 StringBuffer buffer = new StringBuffer(str);  
 buffer.reverse();  
 return buffer.toString();  
 }  
 public String reverseMyStringWithoutBuffer(String str){  
 int length = str.length();  
 String original = str;  
 String reverse = "";  
 for(int i = length-1; i>=0; i--){  
 reverse = reverse + original.charAt(i);   
 }   
 return reverse;  
 }  
 public static void main(String[] args){   
   
 ReverseString rs = new ReverseString();  
 System.out.println("Reversed String using Buffer: "  
                          +rs.reverseMyString("SeleniumWebdriver"));  
 System.out.println("Reversed String without Buffer: "  
                           +rs.reverseMyStringWithoutBuffer("SeleniumWebDriver"));  
 }  
  
  
}

5.

### How to find the number is prime or not by userinput

package Test;  
  
import java.util.Scanner;  
  
public class PrimeNumbers {  
  
  
    public static void main(String[] args) {  
       Scanner s = new Scanner(System.in);  
        System.out.print("Enter a number : ");  
        int n = s.nextInt();  
        if (isPrime(n)) {  
            System.out.println(n + " is a prime number");  
        } else {  
            System.out.println(n + " is not a prime number");  
        }  
    }  
  
    public static boolean isPrime(int n) {  
        if (n <= 1) {  
            return false;  
        }  
        for (int i = 2; i < Math.sqrt(n); i++) {  
            if (n % i == 0) {  
                return false;  
            }  
        }  
        return true;  
    }  
}  
  
  
**Output:**  
Enter a number : 56  
56 is not a prime number  
  
Enter a number : 19  
19 is a prime number

6.

### Palindrome Example in java

package Test;  
  
import java.io.BufferedReader;  
import java.io.InputStreamReader;  
  
public class Palindrome  {  
   public static void main(String [] args){  
   try{  
   BufferedReader object = new BufferedReader(new InputStreamReader(System.in));  
   System.out.println("Enter number");  
   int num= Integer.parseInt(object.readLine());  
   int n = num;  
   int rev=0;  
   System.out.println("Number: ");  
   System.out.println(" "+ num);  
   for (int i=0; i<=num; i++){  
   int r=num%10;  
   num=num/10;  
   rev=rev\*10+r;  
   i=0;  
   }  
   System.out.println("After reversing the number: "+ " ");  
   System.out.println(" "+ rev);   
   if(n == rev){  
   System.out.print("Number is palindrome!");  
   }  
   else{  
   System.out.println("Number is not palindrome!");  
   }  
   }  
   catch(Exception e){  
   System.out.println("Out of range!");  
   }  
   }  
 }

**Output:**

1.

Number:

 231

After reversing the number:

 132

Number is not palindrome!

2.Enter number

252

Number:

 252

After reversing the number:

 252

Number is palindrome!

7.

### Two Maximum numbers

package Test;  
  
public class TwoMaximumNumbers {  
 public void printTwoMaximumNumbers(int[] nums){  
   
 int maxOne = 0;  
 int maxTwo = 0;  
 for(int n:nums){  
 if(maxOne<n){  
 maxTwo = maxOne;  
 maxOne = n;  
 }else if(maxTwo<n){  
 maxTwo = n;  
 }  
   
 }  
   
 System.out.println("Maximum number one"+maxOne);  
 System.out.println("Maximum number two"+ maxTwo);  
 }  
  
  
 public static void main(String []args){  
   
 int num[] = {5,34,78,2,45,1,99,23};  
        TwoMaximumNumbers tmn = new TwoMaximumNumbers();  
        tmn.printTwoMaximumNumbers(num);  
 }  
}

**Output:**

Maximum number one99

Maximum number two78

8.

### Find the Longest Substring in the given String Without Repetition of characters

Given a string, find the longest substrings without repeating characters. Iterate through the given string, find the longest maximum substrings.  
package Logs;  
import java.util.HashSet;  
import java.util.Set;  
  
public class longestsubstring {  
  
 private Set<String> subStrList = new HashSet<String>();  
 private int finalSubStrSize = 0;  
   
 public Set<String> getLongestSubstr(String input){  
 //reset instance variables  
 subStrList.clear();  
 finalSubStrSize = 0;  
 // have a boolean flag on each character ascii value  
 boolean[] flag = new boolean[256];  
 int j = 0;  
 char[] inputCharArr = input.toCharArray();  
 for (int i = 0; i < inputCharArr.length; i++) {  
 char c = inputCharArr[i];  
 if (flag[c]) {  
 extractSubString(inputCharArr,j,i);  
 for (int k = j; k < i; k++) {  
 if (inputCharArr[k] == c) {  
 j = k + 1;  
 break;  
 }  
 flag[inputCharArr[k]] = false;  
 }   
 } else {  
 flag[c] = true;  
 }  
 }  
 extractSubString(inputCharArr,j,inputCharArr.length);  
 return subStrList;  
 }  
   
 private String extractSubString(char[] inputArr, int start, int end){  
   
 StringBuilder sb = new StringBuilder();  
 for(int i=start;i<end;i++){  
 sb.append(inputArr[i]);  
 }  
 String subStr = sb.toString();  
 if(subStr.length() > finalSubStrSize){  
 finalSubStrSize = subStr.length();  
 subStrList.clear();  
 subStrList.add(subStr);  
 } else if(subStr.length() == finalSubStrSize){  
 subStrList.add(subStr);  
 }  
   
 return sb.toString();  
 }  
  
 public static void main(String a[]){  
 longestsubstring mls = new longestsubstring();  
 System.out.println(mls.getLongestSubstr("automation\_tester"));  
 System.out.println(mls.getLongestSubstr("Selenium\_Webdriver\_is\_AutomationTesting"));  
 System.out.println(mls.getLongestSubstr("sdsjk\_uippj\_agjhs\_7890"));  
 System.out.println(mls.getLongestSubstr("abcabcbb"));  
 }  
}  
  
  
**OutPut:**  
[ion\_tes, mation\_]  
[nium\_Webdr, um\_Webdriv]  
[agjhs\_7890]  
[cab, abc, bca]

9.

### Swapping of Two Numbers without Using Temp Variable

package Logs;  
  
public class Swapping {  
  
    public static void main(String a[]){  
        int x = 10;  
        int y = 20;  
        System.out.println("Before swap:");  
        System.out.println("x value: "+x);  
        System.out.println("y value: "+y);  
        x = x+y;  
        y=x-y;  
        x=x-y;  
        System.out.println("After swap:");  
        System.out.println("x value: "+x);  
        System.out.println("y value: "+y);  
    }  
}  
 **OutPut:**  
 **Before swap:**  
x value: 10  
y value: 20  
**After swap:**  
x value: 20  
  
y value: 10

**10.How can you remove the multiple spaces of String**  
  
package Logs;  
import java.util.StringTokenizer;  
  
public class Mutiplespaces {  
    public static void main(String a[]){  
        String str = "Selenium    Webdriver Automation      Testing";  
        StringTokenizer st = new StringTokenizer(str, " ");  
        StringBuffer sb = new StringBuffer();  
        while(st.hasMoreElements()){  
            sb.append(st.nextElement()).append(" ");  
        }  
        System.out.println(sb.toString().trim());  
    }  
}