




Java Program: questions and answers

1. How to convert Integer to String in Java?




```
Main.java   Run
```

```
1 //package softwareTestingMaterial;
2
3 public class STM {
4
5     public static void main(String[] args) {
6         int x = 123;
7         int y = 456;
8         String s1 = Integer.toString(x);
9         String s2 = Integer.toString(y);
10        System.out.println(s1);
11        System.out.println(s2);
12    }
13 }
```

2. How to convert String to Integer in Java?

Main.java	  Run	Output
<pre>1 //package softwareTestingMaterial; 2 3 public class STM { 4 5 public static void main(String[] args) { 6 String str = "100"; 7 // Integer.parseInt() 8 System.out.println(Integer.parseInt(str)); 9 } 10 } 11</pre>		<pre>java -cp /tmp/Tcy0iG6zhr/STM 100 === Code Execution Successful ===</pre>



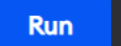
3. How to convert Char to Integer in Java?

```
Main.java   
1 //package softwareTestingMaterial;
2
3 public class STM {
4
5     public static void main(String[] args) {
6         // Initializing a character(ch)
7         char c = '9';
8         // Converting the character to an interger value
9         int number = Integer.parseInt(String.valueOf(c));
10        System.out.println(number);
11    }
12 }
13
```

4. Write a program to print the pattern given below

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```




Here is the program to print the pattern mentioned above

```
Main.java   
1 public class NumberPattern {
2
3     public static void main(String[] args) {
4         for (int x = 1; x <= 5; x++) {
5             for (int y = 1; y <= x; y++) {
6                 System.out.print(y+" ");
7             }
8             System.out.println();
9         }
10    }
11 }
12
```

5. Write a program to print the pattern given below (Left Triangle Star Pattern)

```
*  
* *  
* * *  
* * * *  
* * * * *
```




Here is the program to print the pattern mentioned above

```
Main.java   
1 public class star{
2     public static void main(String args[]) {
3         int x, y, row=5;
4         for(x=0; x<=row; x++) {
5             for(y=0; y<=x; y++) {
6                 System.out.print("* ");
7             }
8             //Cursor goes to the new line after printing each line.
9             System.out.println();
10        }
11    }
12 }
```

6. Write a program to print the pattern given below (Right Triangle Star Pattern)

```
*  
* *  
* * *  
* * * *  
* * * * *
```

Here is the program to print the pattern mentioned above



```
Main.java   
```

```
1 public class star{  
2     public static void main(String args[]) {  
3         //x for rows,y for columns, and row denotes the number of rowstoprint  
4         int x, y, row=5;  
5         //outer loop for number of rows  
6         for(x=0; x<row; x++) { //inner loop for columns  
7             for(y=2*(row-x); y>=0; y--) {  
8                 //To prints spaces  
9                 System.out.print(" ");  
10            }  
11            //Inner loop for columns  
12            for(y=0; y<=x; y++){  
13                //To prints stars  
14                System.out.print("* ");  
15            }  
16            //Cursor goes to the new line after printing each line  
17            System.out.println();  
18        }  
19    }  
20 }  
21
```



7. Write a program to print the pattern given below (Pyramid Star Pattern)

```
*  
* *  
* * *  
* * * *  
* * * * *
```


Here is the program to print the pattern mentioned above

```
Main.java   Run  
1 public class pyramid{  
2 public static void main(String args[]) {  
3 //x for rows, y for columns, and row denotes the number of rows to print  
4 int x, y, row = 5;  
5 //Outer loop for rows  
6 for (x=0; x<row; x++) { //inner loop for space  
7 for (y=row-x; y>1; y--) { //To print space between two stars  
8 System.out.print(" ");  
9 }  
10 //inner loop for columns  
11 for (y=0; y<=x; y++ ) {  
12 //To print star  
13 System.out.print("* ");  
14 } //Cursor goes to the new line after printing each line.  
15 System.out.println();  
16 }  
17 }  
18 }
```




8. How to reverse a String in Java?

```
Main.java   Run  
1 public class ReverseString {  
2 public static void main(String[] args) {  
3 // Using StringBuffer class  
4 StringBuffer a = new StringBuffer("Software Testing Material");  
5 // use reverse() method to reverse string  
6 System.out.println(a.reverse());  
7 }  
8 }
```

Another method:


```
Main.java   
1 public class ReverseString {
2     public static void main(String[] args) {
3         String input="Software Testing Material";
4         StringBuilder input1 = new StringBuilder();
5         input1.append(input);
6         input1=input1.reverse();
7         for (int i=0;i<input1.length();i++)
8             System.out.print(input1.charAt(i));
9     }
10 }
```

9. How To Find The Largest Value From The Given Array.

```
Main.java   
1 public class LargestValue {
2     public static void main(String[] args){
3         int[] arr={28,3,15,9,17,4,23,2};
4         int val=arr[0];
5         for(int i=0; i<arr.length; i++){
6             if(arr[i] > val){
7                 val=arr[i];
8             }
9         }
10        System.out.println("Largest value in the Given Array is "+ val);
11    }
12 }
```

10. How to display all the prime numbers between 1 and 100

The number which is only divisible by 1 and itself is known as a prime number. For example 2, 3, 5, 7, 11... are prime numbers.

```
Main.java   
1 public class PrimeNumbersOneToHundred {
2     public static void main (String[] args){
3         int i =0;
4         int num =0;
5         String primeNumbers = "";
6         for (i = 1; i <= 100; i++){
7             int counter=0;
8             for(num =i; num>=1; num--){
9                 if(i%num==0){
10                     counter = counter + 1;
11                 }
12             }
13             if (counter ==2){
14                 primeNumbers = primeNumbers + i + " ";
15             }
16         }
17         System.out.println("Prime numbers from 1 to 100 are :");
18         System.out.println(primeNumbers);
19     }
20 }
```

11. How to display all the prime numbers between 1 and n (n is the number, get the input from user)

Main.java

```
1 import java.util.Scanner;
2 public class PrimeNumbersOneToN {
3     public static void main (String[] args){
4         Scanner scanner = new Scanner(System.in);
5         int i =0;
6         int num =0;
7         String primeNumbers = "";
8         System.out.println("Enter the value of n :");
9         int n = scanner.nextInt();
10        scanner.close();
11        for (i = 1; i <= n; i++)
12        {
13            int counter=0;
14            for(num =i; num>=1; num--)
15            {
16                if(i%num==0)
17                {
18                    counter = counter + 1;
19                }
20            }
21            if (counter ==2)
22            {
23                primeNumbers = primeNumbers + i + " ";
24            }
25        }
26        System.out.println("Prime numbers from 1 to n are :");
27        System.out.println(primeNumbers);
28    }
29 }
30
```

12. How to find the given number is a prime number or not by getting input from the user

Main.java

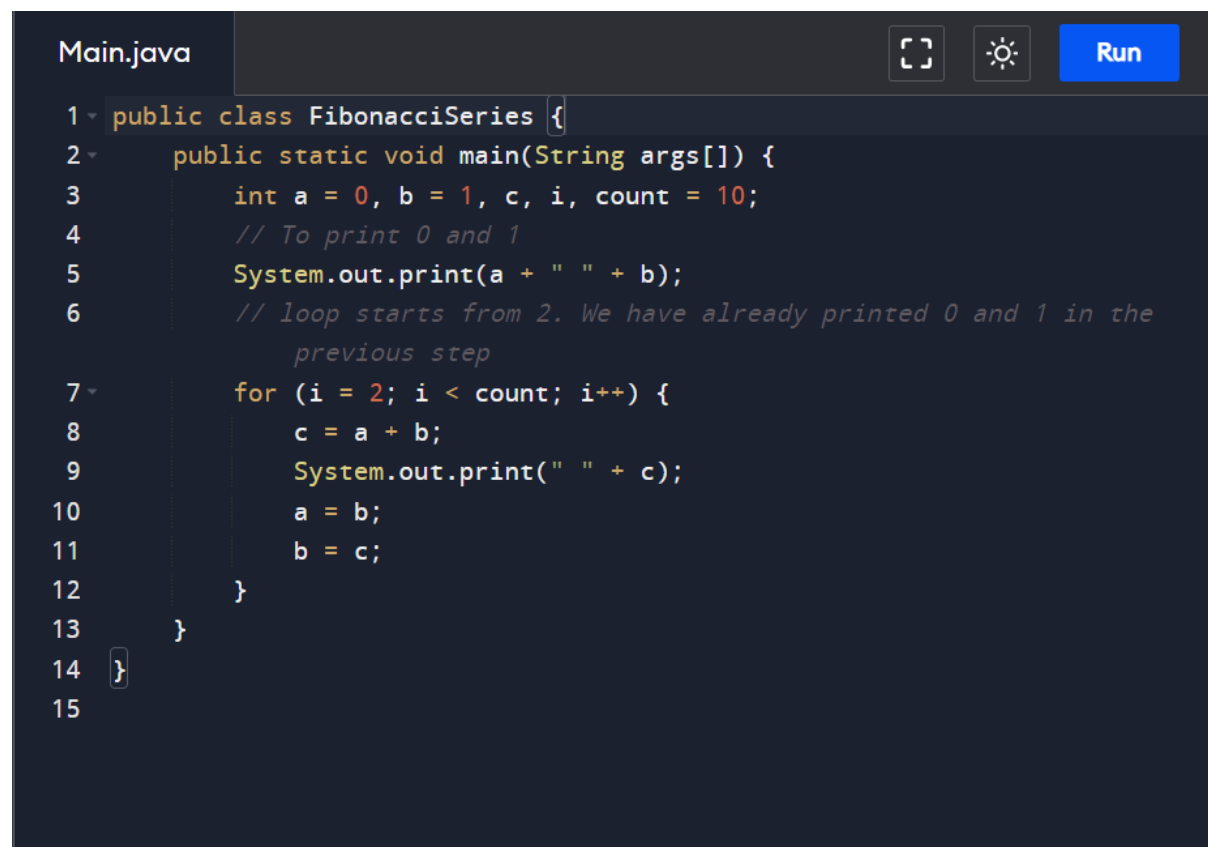


Run

```
1 import java.util.Scanner;
2 public class PrimeNumberVerification {
3     public static void main(String args[])
4     {
5         int i, j, flag = 0;
6         System.out.print("Enter any number which you want to verify whether it is a prime number or not :");
7         Scanner s = new Scanner(System.in);
8         j = s.nextInt();
9         for( i = 2; i < j; i++){
10             if(j % i == 0){
11                 flag = 0;
12                 break;
13             }
14             else
15             {
16                 flag = 1;
17             }
18         }
19         if(flag == 1){
20             System.out.println(j+" is a prime number.");
21         }
22         else{
23             System.out.println(j+" is not a prime number.");
24         }
25     }
26 }
27
```

13. Write a program to print Fibonacci Series



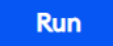
Method 1:



The screenshot shows a Java IDE with a file named 'Main.java'. The code defines a class 'FibonacciSeries' with a 'main' method. The 'main' method initializes variables 'a' (0), 'b' (1), 'c', 'i', and 'count' (10). It prints 'a' and 'b' (0 and 1). Then, it enters a 'for' loop starting from 'i = 2' to 'count - 1'. Inside the loop, it calculates 'c = a + b', prints 'c', and updates 'a = b' and 'b = c'. The output of the program is '0 1 1 2 3 5 8 13 21 34'.

```
1 public class FibonacciSeries {
2     public static void main(String args[]) {
3         int a = 0, b = 1, c, i, count = 10;
4         // To print 0 and 1
5         System.out.print(a + " " + b);
6         // loop starts from 2. We have already printed 0 and 1 in the
           previous step
7         for (i = 2; i < count; i++) {
8             c = a + b;
9             System.out.print(" " + c);
10            a = b;
11            b = c;
12        }
13    }
14 }
15
```

Method 2:

```
Main.java   
1 import java.util.Scanner;
2 public class FibonacciSeriesOne {
3     public static void main(String[] args){
4         System.out.println("Enter Iteration to print Fibonacci Series");
5         FibonacciCheck.checkFibonacci(new Scanner(System.in).nextInt());
6     }
7 }
8 class FibonacciCheck {
9     public static void checkFibonacci(int number){
10         int first=0,second=1;
11         int third=0;
12         int i=1;
13         System.out.print("Fibonacci Series upto: "+number+" is ");
14         System.out.print(first+","+second+",");
15         while(i<=number){
16             third=first+second;
17             System.out.print(third+",");
18             first=second;
19             second=third;
20             ++i;
21         }
22     }
23 }
```

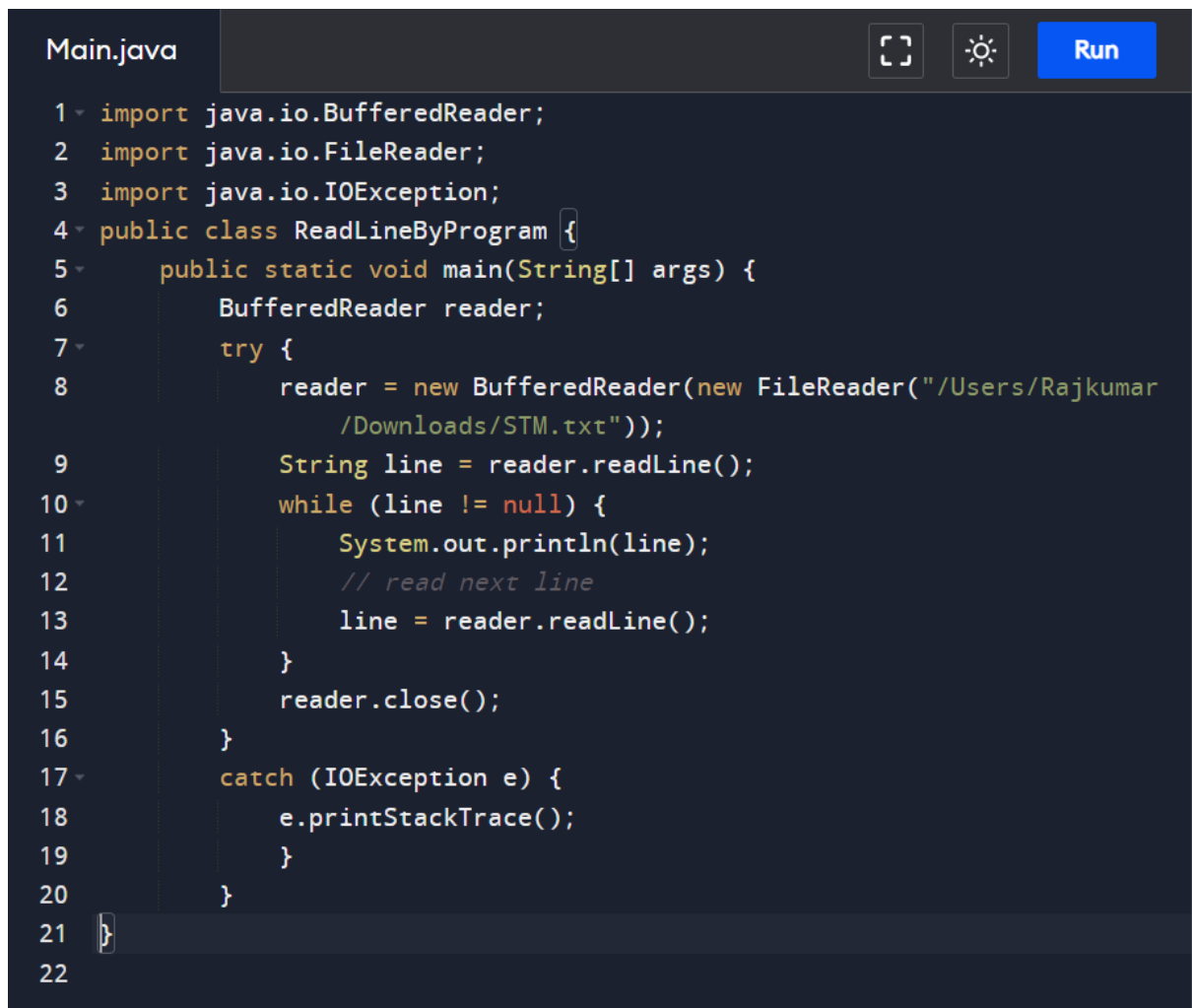
14. How to read a file line by line in Java?



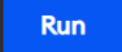
We can read a file line by line in Java in two ways.

1. BufferedReader Class
2. Scanner Class

Using BufferedReader Class:

BufferedReader Class belongs to java.io package and it provides readLine() method to read a file line by line in Java.



```
Main.java   
```

```
1 import java.io.BufferedReader;
2 import java.io.FileReader;
3 import java.io.IOException;
4 public class ReadLineByProgram {
5     public static void main(String[] args) {
6         BufferedReader reader;
7         try {
8             reader = new BufferedReader(new FileReader("/Users/Rajkumar
9                 /Downloads/STM.txt"));
10            String line = reader.readLine();
11            while (line != null) {
12                System.out.println(line);
13                // read next line
14                line = reader.readLine();
15            }
16            reader.close();
17        } catch (IOException e) {
18            e.printStackTrace();
19        }
20    }
21 }
22
```

Using Scanner Class:

Java Scanner class provides the `nextLine()` method to facilitates line by line of file's content.

Main.java

Run

```
1 import java.io.File;
2 import java.io.FileNotFoundException;
3 import java.util.Scanner;
4 public class ReadLineByProgram {
5     public static void main(String[] args) {
6         try {
7             Scanner scanner = new Scanner(new File("/Users/Rajkumar
              /Downloads/STM.txt"));
8             while (scanner.hasNextLine()) {
9                 System.out.println(scanner.nextLine());
10            }
11            scanner.close();
12        }
13        catch (FileNotFoundException e) {
14            e.printStackTrace();
15        }
16    }
17 }
```



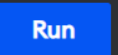
15. Swap string without 3rd variable?

Main.java	<div><div></div><div></div><div>Run</div></div>	Output
<pre>1 // Java program to swap two strings without using a temporary 2 // variable. 3 class Swap 4 { 5 public static void main(String args[]) 6 { 7 // Declare two strings 8 String a = "Hello"; 9 String b = "World"; 10 // Print String before swapping 11 System.out.println("Strings before swap: a = " + 12 a + " and b = "+b); 13 // append 2nd string to 1st 14 a = a + b; 15 b=a.substring(0,a.length()-b.length()); 16 a=a.substring(b.length()); 17 // print String after swapping 18 System.out.println("Strings after swap: a = " + 19 a + " and b = " + b); 20 } 21 } 22</pre>	<pre>java -cp /tmp/akaq080wYX/Swap Strings before swap: a = Hello and b = World Strings after swap: a = World and b = Hello === Code Execution Successful ===</pre>	




16.Duplicates in a String?

```
Main.java   
1 public class FindDuplicates {
2     public static void main(String[] args) {
3         String str = "geeksforgeeks";
4         char[] chars = str.toCharArray();
5
6         System.out.println("Duplicate characters:");
7         for (int i = 0; i < chars.length; i++) {
8             for (int j = i + 1; j < chars.length; j++) {
9                 if (chars[i] == chars[j]) {
10                     System.out.print(chars[j] + " ");
11                 }
12             }
13         }
14     }
15 }
16
```




17. How to find the length of the string without using length?

```
Main.java   
1 import java.util.Scanner;
2 public class Main {
3     public static void main(String[] args) {
4         int length=0;
5         String s = "prepinsta";
6         for (char c1 : s.toCharArray())
7             length++;
8         System.out.println("Length of String is : "+length);
9     }
10 }
```

18.Largest number in an Array

```
Main.java   
1 import java.util.Scanner;
2 public class Main
3 {
4     public static void main(String args[])
5     {
6         int arr[] = {12, 13, 1, 10, 34, 10};
7         int max = arr[0];
8         for(int i=0; i<arr.length; i++)
9         {
10             if(max < arr[i])
11             {
12                 max = arr[i];
13             }
14         }
15         System.out.print(max);
16     }
17 }
```

19.Reverse a string without using reverse function

```
Main.java   
1 // java program to reverse a word
2 import java.io.*;
3 import java.util.Scanner;
4 class GFG {
5     public static void main (String[] args) {
6         String str= "Geeks", nstr="";
7         char ch;
8         System.out.print("Original word: ");
9         System.out.println("Geeks"); //Example word
10        for (int i=0; i<str.length(); i++)
11        {
12            ch= str.charAt(i); //extracts each character
13            nstr= ch+nstr; //adds each character in front of the existing string
14        }
15        System.out.println("Reversed word: "+ nstr);
16    }
17 }
```

20. Write code to print only the even numbers from an array.

```
Main.java
1 public class EvenNumbersFromArray {
2     public static void main(String[] args) {
3         int[] numbers = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
4
5         System.out.println("Even numbers:");
6         for (int num : numbers) {
7             if (num % 2 == 0) {
8                 System.out.println(num);
9             }
10        }
11    }
12 }
13
```

21. Write code to find special character, number, capital and small letter in a given string.

```
Main.java
1 import java.util.regex.Pattern;
2 import java.util.regex.Matcher;
3 public class CharacterChecker {
4     public static void main(String[] args) {
5         String str = "#GeeksForGeeks123@";
6         // Create a regular expression to match the criteria
7         String regex = "^(?=.*[a-z])(?=.*[A-Z])(?=.*\\d)(?=.*[~!@#$%^&*.,?]).+$";
8         // Check if the string matches the regular expression
9         if (str.matches(regex)) {
10             System.out.println("Yes, the string contains uppercase letters, lowercase letters, special characters, and numeric values.");
11         } else {
12             System.out.println("No, the string does not meet all the criteria.");
13         }
14     }
15 }
```

21. Write code to check if a number is palindrome?


```
Main.java
1 class Main {
2     public static void main(String[] args) {
3         String str = "Radar", reverseStr = "";
4         int strLength = str.length();
5         for (int i = (strLength - 1); i >= 0; --i) {
6             reverseStr = reverseStr + str.charAt(i);
7         }
8         if (str.toLowerCase().equals(reverseStr.toLowerCase())) {
9             System.out.println(str + " is a Palindrome String.");
10        }
11        else {
12            System.out.println(str + " is not a Palindrome String.");
13        }
14    }
15 }
```

22. Write a Java code to identify, if the pair of strings are an Anagram or not

```
Main.java
1 import java.util.Arrays;
2 public class AnagramChecker {
3     public static void main(String[] args) {
4         String string1 = "listen";
5         String string2 = "silent";
6
7         if (isAnagramSort(string1, string2)) {
8             System.out.println("The strings are anagrams.");
9         } else {
10            System.out.println("The strings are not anagrams.");
11        }
12    }
13    static boolean isAnagramSort(String s1, String s2) {
14        if (s1.length() != s2.length()) {
15            return false;
16        }
17        char[] a1 = s1.toCharArray();
18        char[] a2 = s2.toCharArray();
19        Arrays.sort(a1);
20        Arrays.sort(a2);
21        return Arrays.equals(a1, a2);
22    }
23 }
```

23. Write a java code for factorial of a Number using recursive method ?

Java



```
public class Factorial {
    public static void main(String[] args) {
        int number = 5; // Change this to calculate factorial of different numbers
        int factorial = calculateFactorial(number);
        System.out.println("Factorial of " + number + " = " + factorial);
    }

    /**
     * Recursive function to calculate factorial
     * @param n number to calculate factorial
     * @return factorial of n
     */
    public static int calculateFactorial(int n) {
        if (n == 0 || n == 1) {
            // base case: factorial of 0 or 1 is 1
            return 1;
        } else {
            // recursive case: n! = n * (n-1)!
            return n * calculateFactorial(n - 1);
        }
    }
}
```

24. Write Java code to print all the array elements that appear at least 2 times (meaning 2 or greater than two).

```
import java.util.HashMap;
import java.util.Map;

public class DuplicateElements {
    public static void findDuplicates(int[] arr) {
        Map<Integer, Integer> elementCounts = new HashMap<>();

        // Count occurrences of each element
        for (int num : arr) {
            elementCounts.put(num, elementCounts.getOrDefault(num, 0) + 1);
        }

        System.out.println("Duplicate elements:");
        // Print elements that appear more than once
        for (Map.Entry<Integer, Integer> entry : elementCounts.entrySet()) {
            if (entry.getValue() >= 2) {
                System.out.print(entry.getKey() + " ");
            }
        }
        System.out.println(); // New line for better formatting
    }

    public static void main(String[] args) {
        int[] array = {1, 2, 3, 2, 4, 1, 5, 2, 6, 3};
        findDuplicates(array); // Output: Duplicate elements: 1 2 3
        int[] array2 = {1,1,1,1,1,1,1,1};
        findDuplicates(array2); // Output: Duplicate elements: 1
        int[] array3 = {1,2,3,4,5,6,7,8};
        findDuplicates(array3); // Output: Duplicate elements:
    }
}
```

2. Using Nested Loops (Less Efficient, but simpler to understand):

Java



```
public class DuplicateElementsNestedLoop {
    public static void findDuplicates(int[] arr) {
        System.out.println("Duplicate elements:");
        for (int i = 0; i < arr.length; i++) {
            for (int j = i + 1; j < arr.length; j++) {
                if (arr[i] == arr[j]) {
                    System.out.print(arr[i] + " ");
                    // To avoid printing the same duplicate multiple times
                    // you could use a Set to store already printed duplicates or
                    break; // break is added to print each duplicate only once.
                }
            }
        }
        System.out.println();
    }

    public static void main(String[] args) {
        int[] array = {1, 2, 3, 2, 4, 1, 5, 2, 6, 3};
        findDuplicates(array); // Output: Duplicate elements: 1 2 3
        int[] array2 = {1,1,1,1,1,1,1,1};
        findDuplicates(array2); // Output: Duplicate elements: 1
        int[] array3 = {1,2,3,4,5,6,7,8};
        findDuplicates(array3); // Output: Duplicate elements:
    }
}
```