

# Java 8 Interview Sample Coding Questions

---

 [javaconceptoftheday.com/java-8-interview-sample-coding-questions](https://javaconceptoftheday.com/java-8-interview-sample-coding-questions)

pramodbablad

June 26, 2023

---

Here are the some Java 8 interview sample coding questions with answers. I hope it will be helpful for you guys while preparing for an interview.

Java 8 Interview Sample Coding Questions		Java Concept Of The Day
<b>Separate Odd And Even Numbers</b>		<b>Remove Duplicate Elements From List</b>
<pre>listOfIntegers.stream()     .collect(Collectors.partitioningBy(i -&gt; i % 2 == 0));</pre>		<pre>listOfStrings.stream().distinct().collect(Collectors.toList());</pre>
<b>Frequency Of Each Character In String</b>		<b>Frequency Of Each Element In An Array</b>
<pre>inputString.chars()     .mapToObj(c -&gt; (char) c)     .collect(Collectors.groupingBy(Function.identity(),         Collectors.counting()));</pre>		<pre>anyList.stream().collect(Collectors.groupingBy(Function.identity(),     Collectors.counting()));</pre>
<b>Sort The List In Reverse Order</b>		<b>Join List Of Strings With Prefix, Suffix And Delimiter</b>
<pre>anyList.stream().sorted(Comparator.reverseOrder()).forEach(System.out::println);</pre>		<pre>listOfStrings.stream().collect(Collectors.joining("Delimiter", "Prefix",     "Suffix"));</pre>
<b>Print Multiples Of 5 From The List</b>		<b>Maximum &amp; Minimum In A List</b>
<pre>listOfIntegers.stream()     .filter(i -&gt; i % 5 == 0).forEach(System.out::println);</pre>		<pre>listOfIntegers.stream().max(Comparator.naturalOrder()).get(); listOfIntegers.stream().min(Comparator.naturalOrder()).get();</pre>
<b>Merge Two Unsorted Arrays Into Single Sorted Array</b>		<b>Anagram Program In Java 8</b>
<pre>IntStream.concat(Arrays.stream(a), Arrays.stream(b))     .sorted().toArray();</pre>		<pre>s1=Stream.of(s1.split("")).map(String::toUpperCase).sorted().collect     (Collectors.joining()); s2=Stream.of(s2.split("")).map(String::toUpperCase).sorted().collect     (Collectors.joining()); If s1 and s2 are equal, then they are anagrams.</pre>
<b>Merge Two Unsorted Arrays Into Single Sorted Array Without Duplicates</b>		<b>Sum Of All Digits Of A Number</b>
<pre>IntStream.concat(Arrays.stream(a), Arrays.stream(b))     .sorted().distinct().toArray();</pre>		<pre>Stream.of(String.valueOf(inputNumber).split(""))     .collect(Collectors.summingInt(Integer::parseInt));</pre>
<b>Three Max &amp; Min Numbers From The List</b>		<b>Second Largest Number In An Integer Array</b>
<pre>//Min 3 Numbers listOfIntegers.stream().sorted().limit(3).forEach(System.out::println);  //Max 3 Numbers listOfIntegers.stream().sorted(Comparator.reverseOrder()).limit(3).fo rEach(System.out::println);</pre>		<pre>listOfIntegers.stream().sorted(Comparator.reverseOrder()).skip(1)     .findFirst().get();</pre>
<b>Sort List Of Strings In Increasing Order Of Their Length</b>		<b>Common Elements Between Two Arrays</b>
<pre>listOfStrings.stream().sorted(Comparator.comparing(String::length))     .forEach(System.out::println);</pre>		<pre>list1.stream().filter(list2::contains).forEach(System.out::println);</pre>
<b>Sum &amp; Average Of All Elements Of An Array</b>		<b>Reverse Each Word Of A String</b>
<pre>//Sum Arrays.stream(inputArray).sum();  //Average Arrays.stream(inputArray).average().getAsDouble();</pre>		<pre>Arrays.stream(str.split(" "))     .map(word -&gt; new StringBuffer(word).reverse())     .collect(Collectors.joining(" "));</pre>
<b>Reverse An Integer Array</b>		<b>Sum Of First 10 Natural Numbers</b>
<pre>IntStream.rangeClosed(1, array.length)     .map(i -&gt; array[array.length - i])     .toArray();</pre>		<pre>IntStream.range(1, 11).sum();</pre>
<b>Palindrome Program In Java 8</b>		<b>Find Strings Which Start With Number</b>
<pre>IntStream.range(0, str.length()/2)     .noneMatch(i -&gt; str.charAt(i) != str.charAt(str.length() - i - 1));</pre>		<pre>listOfStrings.stream()     .filter(str -&gt; Character.isDigit(str.charAt(0)))     .forEach(System.out::println);</pre>
<b>Last Element Of An Array</b>		<b>Find Duplicate Elements From An Array</b>
<pre>listOfStrings.stream().skip(listOfStrings.size()-1).findFirst().get();</pre>		<pre>listOfIntegers.stream()     .filter(i -&gt; ! set.add(i))     .collect(Collectors.toSet());</pre>
<b>Age Of Person In Years</b>		<b>Fibonacci Series</b>
<pre>LocalDate birthDay = LocalDate.of(1985, 01, 23); LocalDate today = LocalDate.now(); System.out.println(ChronoUnit.YEARS.between(birthDay, today));</pre>		<pre>Stream.iterate(new int[] {0, 1}, f -&gt; new int[] {f[1], f[0]+f[1]})     .limit(10)     .map(f -&gt; f[0])     .forEach(i -&gt; System.out.print(i+" "));</pre>

## Java 8 Interview Coding Questions And Answers :

1) Given a list of integers, separate odd and even numbers?

```

1  import java.util.Arrays;
2  import java.util.List;
3  import java.util.Map;
4  import java.util.Map.Entry;
5  import java.util.Set;
6  import java.util.stream.Collectors;
7  public class Java8Code
8  {
9  public static void main(String[] args)
10 {
11 List<Integer> listOfIntegers = Arrays.asList(71, 18, 42, 21, 67, 32, 95, 14, 56, 87);
12
13 Map<Boolean, List<Integer>> oddEvenNumbersMap =
14 listOfIntegers.stream().collect(Collectors.partitioningBy(i -> i % 2 == 0));
15
16 Set<Entry<Boolean, List<Integer>>> entrySet = oddEvenNumbersMap.entrySet();
17
18 for (Entry<Boolean, List<Integer>> entry : entrySet)
19 {
20 System.out.println("-----");
21
22 if (entry.getKey())
23 {
24 System.out.println("Even Numbers");
25 }
26 else
27 {
28 System.out.println("Odd Numbers");
29 }
30
31 System.out.println("-----");
32
33 List<Integer> list = entry.getValue();
34
35 for (int i : list)
36 {
37 System.out.println(i);
38 }
39 }
40
41
42

```

#### Output :

---

Odd Numbers

---

71  
21  
67  
95  
87

---

Even Numbers

---

18  
42  
32  
14  
56

## 2) How do you remove duplicate elements from a list using Java 8 streams?

```
1  import java.util.Arrays;
2  import java.util.List;
3  import java.util.stream.Collectors;
4  public class Java8Code
5  {
6  public static void main(String[] args)
7  {
8  List<String> listOfStrings = Arrays.asList("Java", "Python", "C#", "Java", "Kotlin",
9  "Python");
10 List<String> uniqueStrngs = listOfStrings.stream().distinct().collect(Collectors.toList());
11
12 System.out.println(uniqueStrngs);
13 }
14 }
15
```

### Output :

[Java, Python, C#, Kotlin]

## 3) How do you find frequency of each character in a string using Java 8 streams?

```
1  import java.util.Map;
2  import java.util.function.Function;
3  import java.util.stream.Collectors;
4  public class Java8Code
5  {
6  public static void main(String[] args)
7  {
8  String inputString = "Java Concept Of The Day";
9
10 Map<Character, Long> charCountMap =
11 inputString.chars()
12 .mapToObj(c -> (char) c)
13 .collect(Collectors.groupingBy(Function.identity(), Collectors.counting()));
14
15 System.out.println(charCountMap);
16 }
17
18
```

### Output :

{=4, a=3, c=1, C=1, D=1, e=2, f=1, h=1, J=1, n=1, O=1, o=1, p=1, T=1, t=1, v=1, y=1}

## 4) How do you find frequency of each element in an array or a list?

```

1  import java.util.Arrays;
2  import java.util.List;
3  import java.util.Map;
4  import java.util.function.Function;
5  import java.util.stream.Collectors;
6  public class Java8Code
7  {
8  public static void main(String[] args)
9  {
10 List<String> stationeryList = Arrays.asList("Pen", "Eraser", "Note Book", "Pen", "Pencil",
11 "Stapler", "Note Book", "Pencil");
12 Map<String, Long> stationeryCountMap =
13 stationeryList.stream().collect(Collectors.groupingBy(Function.identity(),
14 Collectors.counting()));
15
16 System.out.println(stationeryCountMap);
17 }
18 }

```

**Output :**

{Pen=2, Stapler=1, Pencil=2, Note Book=2, Eraser=1}

**5) How do you sort the given list of decimals in reverse order?**

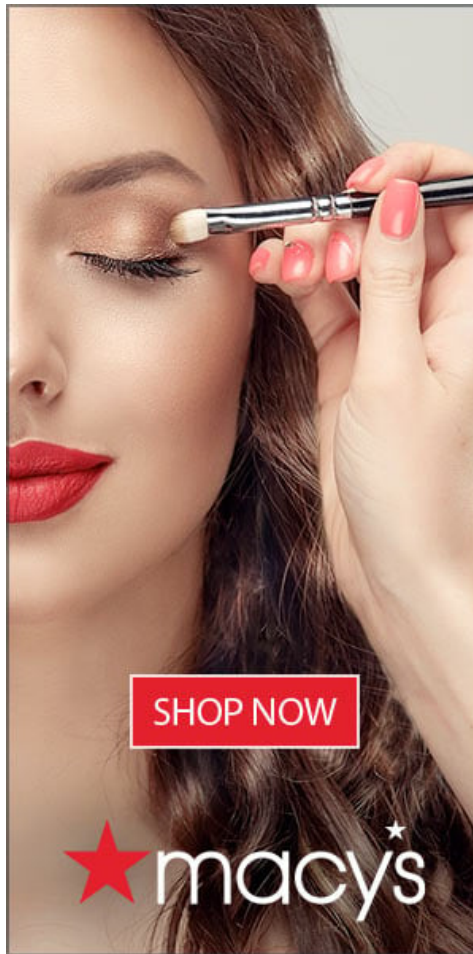
```

1  import java.util.Arrays;
2  import java.util.Comparator;
3  import java.util.List;
4  public class Java8Code
5  {
6  public static void main(String[] args)
7  {
8  List<Double> decimalList = Arrays.asList(12.45, 23.58, 17.13, 42.89, 33.78, 71.85, 56.98,
9  21.12);
10 decimalList.stream().sorted(Comparator.reverseOrder()).forEach(System.out::println);
11 }
12 }
13

```

**Output :**

71.85  
56.98  
42.89  
33.78  
23.58  
21.12  
17.13  
12.45



6) Given a list of strings, join the strings with '[' as prefix, ']' as suffix and ',' as delimiter?

```
1  import java.util.Arrays;
2  import java.util.List;
3  import java.util.stream.Collectors;
4  public class Java8Code
5  {
6  public static void main(String[] args)
7  {
8
9  String joinedString = listOfStrings.stream().collect(Collectors.joining("[", "]", "[" + "," + "]"));
10
11 System.out.println(joinedString);
12 }
13 }
14
15
```

Output :

7) From the given list of integers, print the numbers which are multiples of 5?

```

1  import java.util.Arrays;
2  import java.util.List;
3  public class Java8Code
4  {
5      public static void main(String[] args)
6      {
7          List<Integer> listOfIntegers = Arrays.asList(45, 12, 56, 15, 24, 75, 31, 89);
8
9          listOfIntegers.stream().filter(i -> i % 5 == 0).forEach(System.out::println);
10     }
11 }
12

```

**Output :**

```

45
15
75

```

**8) Given a list of integers, find maximum and minimum of those numbers?**

```

1  import java.util.Arrays;
2  import java.util.Comparator;
3  import java.util.List;
4  public class Java8Code
5  {
6      public static void main(String[] args)
7      {
8          List<Integer> listOfIntegers = Arrays.asList(45, 12, 56, 15, 24, 75, 31, 89);
9
10         int max = listOfIntegers.stream().max(Comparator.naturalOrder()).get();
11         System.out.println("Maximum Element : "+max);
12
13         int min = listOfIntegers.stream().min(Comparator.naturalOrder()).get();
14         System.out.println("Minimum Element : "+min);
15     }
16 }
17
18
19

```

**Output :**

```

Maximum Element : 89
Minimum Element : 12

```

**9) How do you merge two unsorted arrays into single sorted array using Java 8 streams?**

```

1  import java.util.Arrays;
2  import java.util.stream.IntStream;
3  public class Java8Code
4  {
5      public static void main(String[] args)
6      {
7          int[] a = new int[] {4, 2, 7, 1};
8
9          int[] b = new int[] {8, 3, 9, 5};
10
11         int[] c = IntStream.concat(Arrays.stream(a), Arrays.stream(b)).sorted().toArray();
12
13         System.out.println(Arrays.toString(c));
14     }
15 }
16

```

**Output :**

[1, 2, 3, 4, 5, 7, 8, 9]

**10) How do you merge two unsorted arrays into single sorted array without duplicates?**

```

1  import java.util.Arrays;
2  import java.util.stream.IntStream;
3  public class Java8Code
4  {
5      public static void main(String[] args)
6      {
7          int[] a = new int[] {4, 2, 5, 1};
8
9          int[] b = new int[] {8, 1, 9, 5};
10
11         int[] c = IntStream.concat(Arrays.stream(a), Arrays.stream(b)).sorted().distinct().toArray();
12
13         System.out.println(Arrays.toString(c));
14     }
15 }
16

```

**Output :**

[1, 2, 4, 5, 8, 9]

**11) How do you get three maximum numbers and three minimum numbers from the given list of integers?**



```

1  import java.util.Arrays;
2  import java.util.Comparator;
3  import java.util.List;
4  public class Java8Code
5  {
6  public static void main(String[] args)
7  {
8  List<Integer> listOfIntegers = Arrays.asList(45, 12, 56, 15, 24, 75, 31, 89);
9
10
11 System.out.println("-----");
12 System.out.println("Minimum 3 Numbers");
13
14 System.out.println("-----");
15
16 listOfIntegers.stream().sorted().limit(3).forEach(System.out::println);
17
18
19 System.out.println("-----");
20
21 System.out.println("Maximum 3 Numbers");
22
23 System.out.println("-----");
24
25 listOfIntegers.stream().sorted(Comparator.reverseOrder()).limit(3).forEach(System.out::println);
26 }
27 }
28
29
30
31

```

#### Output :

---

Minimum 3 Numbers

---

12  
15  
24

---

Maximum 3 Numbers

---

89  
75  
56

#### 12) Java 8 program to check if two strings are anagrams or not?

```

1  import java.util.stream.Collectors;
2  import java.util.stream.Stream;
3  public class Java8Code
4  {
5      public static void main(String[] args)
6      {
7          String s1 = "RaceCar";
8          String s2 = "CarRace";
9
10         s1 = Stream.of(s1.split("")).map(String::toUpperCase).sorted().collect(Collectors.joining());
11         s2 = Stream.of(s2.split("")).map(String::toUpperCase).sorted().collect(Collectors.joining());
12
13         if (s1.equals(s2))
14         {
15             System.out.println("Two strings are anagrams");
16         }
17         else
18         {
19             System.out.println("Two strings are not anagrams");
20         }
21     }
22 }
23
24

```

**Output :**

Two strings are anagrams

### 13) Find sum of all digits of a number in Java 8?

```

1  import java.util.stream.Collectors;
2  import java.util.stream.Stream;
3  public class Java8Code
4  {
5      public static void main(String[] args)
6      {
7          int i = 15623;
8
9          Integer sumOfDigits =
10         Stream.of(String.valueOf(i).split("")).collect(Collectors.summingInt(Integer::parseInt));
11
12         System.out.println(sumOfDigits);
13     }
14 }

```

**Output :**

17

### 14) Find second largest number in an integer array?

```

1  import java.util.Arrays;
2  import java.util.Comparator;
3  import java.util.List;
4  public class Java8Code
5  {
6  public static void main(String[] args)
7  {
8  List<Integer> listOfIntegers = Arrays.asList(45, 12, 56, 15, 24, 75, 31, 89);
9
10 Integer secondLargestNumber =
11 listOfIntegers.stream().sorted(Comparator.reverseOrder()).skip(1).findFirst().get();
12 System.out.println(secondLargestNumber);
13 }
14 }
15

```

**Output :**

75

**15) Given a list of strings, sort them according to increasing order of their length?**

```

1  import java.util.Arrays;
2  import java.util.Comparator;
3  import java.util.List;
4  public class Java8Code
5  {
6  public static void main(String[] args)
7  {
8  List<String> listOfStrings = Arrays.asList("Java", "Python", "C#", "HTML", "Kotlin", "C++",
9  "COBOL", "C");
10
11 listOfStrings.stream().sorted(Comparator.comparing(String::length)).forEach(System.out::println);
12 }
13

```

**Output :**

C  
C#  
C++  
Java  
HTML  
COBOL  
Python  
Kotlin

**16) Given an integer array, find sum and average of all elements?**

```

1  import java.util.Arrays;
2  public class Java8Code
3  {
4      public static void main(String[] args)
5      {
6          int[] a = new int[] {45, 12, 56, 15, 24, 75, 31, 89};
7
8          int sum = Arrays.stream(a).sum();
9
10         System.out.println("Sum = "+sum);
11
12         double average = Arrays.stream(a).average().getAsDouble();
13
14         System.out.println("Average = "+average);
15     }
16 }
17

```

**Output :**

Sum = 347

Average = 43.375

**17) How do you find common elements between two arrays?**

```

1  import java.util.Arrays;
2  import java.util.List;
3  public class Java8Code
4  {
5      public static void main(String[] args)
6      {
7          List<Integer> list1 = Arrays.asList(71, 21, 34, 89, 56, 28);
8
9          List<Integer> list2 = Arrays.asList(12, 56, 17, 21, 94, 34);
10
11         list1.stream().filter(list2::contains).forEach(System.out::println);
12     }
13 }
14

```

**Output :**

21

34

56

**18) Reverse each word of a string using Java 8 streams?**

```

1  import java.util.Arrays;
2  import java.util.stream.Collectors;
3  public class Java8Code
4  {
5      public static void main(String[] args)
6      {
7          String str = "Java Concept Of The Day";
8
9          String reversedStr = Arrays.stream(str.split(" "))
10             .map(word -> new StringBuffer(word).reverse())
11             .collect(Collectors.joining(" "));
12
13         System.out.println(reversedStr);
14     }
15 }
16

```

**Output :**

avaJ tpecnoC fO ehT yaD

**19) How do you find sum of first 10 natural numbers?**

```

1  import java.util.stream.IntStream;
2  public class Java8Code
3  {
4      public static void main(String[] args)
5      {
6          int sum = IntStream.range(1, 11).sum();
7
8          System.out.println(sum);
9      }
10 }
11

```

**Output :**

55

**20) Reverse an integer array**

```

1  import java.util.Arrays;
2  import java.util.stream.IntStream;
3  public class Java8Code
4  {
5      public static void main(String[] args)
6      {
7          int[] array = new int[] {5, 1, 7, 3, 9, 6};
8
9          int[] reversedArray = IntStream.rangeClosed(1, array.length).map(i -> array[array.length -
10             i]).toArray();
11
12         System.out.println(Arrays.toString(reversedArray));
13     }
14 }

```

**Output :**

[6, 9, 3, 7, 1, 5]

**21) Print first 10 even numbers**

```

1 import java.util.stream.IntStream;
2 public class Java8Code
3 {
4     public static void main(String[] args)
5     {
6         IntStream.rangeClosed(1, 10).map(i -> i * 2).forEach(System.out::println);
7     }
8 }
9

```

**Output :**

```

2
4
6
8
10
12
14
16
18
20

```

**22) How do you find the most repeated element in an array?**

```

1 import java.util.Arrays;
2 import java.util.List;
3 import java.util.Map;
4 import java.util.Map.Entry;
5 import java.util.function.Function;
6 import java.util.stream.Collectors;
7 public class Java8Code
8 {
9     public static void main(String[] args)
10    {
11        List<String> listOfStrings = Arrays.asList("Pen", "Eraser", "Note Book", "Pen", "Pencil",
12        "Pen", "Note Book", "Pencil");
13        Map<String, Long> elementCountMap = listOfStrings.stream()
14        .collect(Collectors.groupingBy(Function.identity(), Collectors.counting()));
15        Entry<String, Long> mostFrequentElement =
16        elementCountMap.entrySet().stream().max(Map.Entry.comparingByValue()).get();
17        System.out.println("Most Frequent Element : "+mostFrequentElement.getKey());
18        System.out.println("Count : "+mostFrequentElement.getValue());
19    }
20 }
21 }
22 }
23 }

```

**Output :**

```

Most Frequent Element : Pen
Count : 3

```

**23) Palindrome program using Java 8 streams**

```

1  import java.util.stream.IntStream;
2  public class Java8Code
3  {
4      public static void main(String[] args)
5      {
6          String str = "ROTATOR";
7
8          boolean isItPalindrome = IntStream.range(0, str.length()/2).
9              noneMatch(i -> str.charAt(i) != str.charAt(str.length() - i - 1));
10
11         if (isItPalindrome)
12         {
13             System.out.println(str+" is a palindrome");
14         }
15         else
16         {
17             System.out.println(str+" is not a palindrome");
18         }
19     }
20 }
21

```

**Output :**

ROTATOR is a palindrome

**24) Given a list of strings, find out those strings which start with a number?**

```

1  import java.util.Arrays;
2  import java.util.List;
3  public class Java8Code
4  {
5      public static void main(String[] args)
6      {
7          List<String> listOfStrings = Arrays.asList("One", "Two", "Three", "Four", "Five", "Six");
8
9          listOfStrings.stream().filter(str ->
10             Character.isDigit(str.charAt(0))).forEach(System.out::println);
11     }
12 }

```

**Output :**

Two  
Three  
Five

**25) How do you extract duplicate elements from an array?**

```

1  import java.util.Arrays;
2  import java.util.HashSet;
3  import java.util.List;
4  import java.util.Set;
5  import java.util.stream.Collectors;
6  public class Java8Code
7  {
8  public static void main(String[] args)
9  {
10 List<Integer> listOfIntegers = Arrays.asList(111, 222, 333, 111, 555, 333, 777, 222);
11
12 Set<Integer> uniqueElements = new HashSet<>();
13
14 Set<Integer> duplicateElements = listOfIntegers.stream().filter(i -> !
15 uniqueElements.add(i)).collect(Collectors.toSet());
16
17 System.out.println(duplicateElements);
18 }
19

```

**Output :**

[333, 222, 111]

## 26) Print duplicate characters in a string?

```

1  import java.util.Arrays;
2  import java.util.HashSet;
3  import java.util.Set;
4  import java.util.stream.Collectors;
5  public class Java8Code
6  {
7  public static void main(String[] args)
8  {
9  String inputString = "Java Concept Of The Day".replaceAll("\\s+", "").toLowerCase();
10
11 Set<String> uniqueChars = new HashSet<>();
12
13 Set<String> duplicateChars =
14 Arrays.stream(inputString.split(""))
15 .filter(ch -> ! uniqueChars.add(ch))
16 .collect(Collectors.toSet());
17
18 System.out.println(duplicateChars);
19 }
20
21

```

**Output :**

[a, c, t, e, o]

## 27) Find first repeated character in a string?



```

1  import java.util.Arrays;
2  import java.util.LinkedHashMap;
3  import java.util.Map;
4  import java.util.function.Function;
5  import java.util.stream.Collectors;
6  public class Java8Code
7  {
8  public static void main(String[] args)
9  {
10 String inputString = "Java Concept Of The Day".replaceAll("\\s+", "").toLowerCase();
11
12 Map<String, Long> charCountMap =
13 Arrays.stream(inputString.split(""))
14 .collect(Collectors.groupingBy(Function.identity(), LinkedHashMap::new,
15 Collectors.counting()));
16
17 String firstRepeatedChar = charCountMap.entrySet()
18 .stream()
19 .filter(entry -> entry.getValue() > 1)
20 .map(entry -> entry.getKey())
21 .findFirst()
22 .get();
23
24 System.out.println(firstRepeatedChar);
25 }
26 }

```

**Output :**

a

**28) Find first non-repeated character in a string?**

```

1  import java.util.Arrays;
2  import java.util.LinkedHashMap;
3  import java.util.Map;
4  import java.util.function.Function;
5  import java.util.stream.Collectors;
6  public class Java8Code
7  {
8  public static void main(String[] args)
9  {
10 String inputString = "Java Concept Of The Day".replaceAll("\\s+", "").toLowerCase();
11
12 Map<String, Long> charCountMap =
13 Arrays.stream(inputString.split(""))
14 .collect(Collectors.groupingBy(Function.identity(), LinkedHashMap::new,
15 Collectors.counting()));
16
17 String firstNonRepeatedChar = charCountMap.entrySet()
18 .stream()
19 .filter(entry -> entry.getValue() == 1)
20 .map(entry -> entry.getKey())
21 .findFirst()
22 .get();
23
24 System.out.println(firstNonRepeatedChar);
25 }
26 }

```

**Output :**

j

### 29) Fibonacci series

```
1  import java.util.stream.Stream;
2  public class Java8Code
3  {
4      public static void main(String[] args)
5      {
6          Stream.iterate(new int[] {0, 1}, f -> new int[] {f[1], f[0]+f[1]})
7              .limit(10)
8              .map(f -> f[0])
9              .forEach(i -> System.out.print(i+" "));
10     }
11 }
12
```

**Output :**

0 1 1 2 3 5 8 13 21 34

### 30) First 10 odd numbers

```
1  import java.util.stream.Stream;
2  public class Java8Code
3  {
4      public static void main(String[] args)
5      {
6          Stream.iterate(new int[] {1, 3}, f -> new int[] {f[1], f[1]+2})
7              .limit(10)
8              .map(f -> f[0])
9              .forEach(i -> System.out.print(i+" "));
10     }
11 }
12
```

**Output :**

1 3 5 7 9 11 13 15 17 19

### 31) How do you get last element of an array?

```
1  import java.util.Arrays;
2  import java.util.List;
3  public class Java8Code
4  {
5      public static void main(String[] args)
6      {
7          List<String> listOfStrings = Arrays.asList("One", "Two", "Three", "Four", "Five", "Six");
8          String lastElement = listOfStrings.stream().skip(listOfStrings.size() - 1).findFirst().get();
9          System.out.println(lastElement);
10     }
11 }
12 }
13 }
14
```

**Output :**

Six

### 32) Find the age of a person in years if the birthday has given?

```
1  import java.time.LocalDate;
2  import java.time.temporal.ChronoUnit;
3  public class Java8Code
4  {
5      public static void main(String[] args)
6      {
7          LocalDate birthDay = LocalDate.of(1985, 01, 23);
8          LocalDate today = LocalDate.now();
9          System.out.println(ChronoUnit.YEARS.between(birthDay, today));
10     }
11 }
12
13
```

**Also Read :**

---