



## 10 Tricky Java 8 Coding Questions (Medium → Advanced, Real Interview Problems)

Here are some of the most frequently asked Java 8 coding interview questions with approach, and **outputs** so that it's easy to follow. 🙌

### ♦ Q1. How do you retrieve the 5th element from a List in Java 8?

```
List<String> list = Arrays.asList("a","b","c","d","e","f");
```

🧑🏫 **Logic:**

```
String res= list.stream().skip(4).findFirst().orElse(null);  
System.out.println(res);
```

✅ **Output: e**

---

### ♦ Q2. Given a String, how do you find the character with the second-highest frequency using Java 8 Streams?

(For example: "abbcccd" → b=2)

🧑🏫 **Logic:**

```
String s = "abbcccd";
```

```
Map<Character, Long> map= s.chars().mapToObj(c -> (char) c)  
                           .collect(Collectors.groupingBy(c -> c,  
                                                           Collectors.counting()));  
  
Character second = map.entrySet().stream()  
    .sorted(Map.Entry.<Character, Long>comparingByValue().reversed())  
    .skip(1).findFirst()  
    .get().getKey();  
System.out.println(second + "=" + map.get(second));
```

✅ **Output: b=2**

♦ **Q3. Given s = "abcd", how do you generate the pattern abbcccdddd using Java 8 Streams?**

 **Logic:**

```
String s = "abcd";

String result = IntStream.range(0, s.length())
    .mapToObj(i -> String.valueOf(s.charAt(i)).repeat(i + 1))
    .collect(Collectors.joining());
System.out.println(result);
```

 **Output:**

**abbcccdddd**

♦ **Q4. How do you sort a list of Employee objects by both name and salary using Java 8?**

 **Logic:**

```
list.stream()
    .sorted(Comparator.comparing(Employee::getName)
        .thenComparing(Employee::getSalary))
    .forEach(System.out::println);
```

♦ **Q5. Given a list of integers, how do you separate the elements into two lists — one containing duplicates and the other containing unique elements using Java 8?**

 **Logic:**

```
List<Integer> list=Arrays.asList(1,2,3,1,2,4,5);
```

```
List<Integer> list = Arrays.asList(1,2,3,1,2,4,5);
```

```
Map<Integer, Long> freq = list.stream()  
    .collect(Collectors.groupingBy(i -> i, Collectors.counting()));
```

```
List<Integer> duplicates=freq.entrySet().stream().filter(e->e.getValue() > 1)  
    .map(Map.Entry::getKey).collect(Collectors.toList());
```

```
List<Integer> unique =  
    freq.entrySet().stream()  
        .filter(e -> e.getValue() == 1)  
        .map(Map.Entry::getKey)  
        .collect(Collectors.toList());
```

 **Output:**

Duplicates → [1, 2]

Unique → [3, 4, 5]

♦ **Q6. How do you find common elements between two sorted integer lists using Java 8 Streams?**

 **Logic:**

```
List<Integer> a = Arrays.asList(1,2,3,4);
```

```
List<Integer> b = Arrays.asList(2,4,6);
```

```
List<Integer> common = a.stream()  
    .filter(b::contains)  
    .collect(Collectors.toList());
```

 **Output:**

[2, 4]

♦ **Q7. From a list of Strings, how do you remove the words that contain any numeric digits using Java 8?**

(Example: "ab1c" should be removed (here number contains in word))

🧑 **Logic:**

```
List<String> list=Arrays.asList("abc", "ab1c", "hello", "h3i");

List<String> result = list.stream()
    .filter(s -> !s.matches(".*\\d.*"))
    .collect(Collectors.toList());
```

✅ **Output:**

```
["abc", "hello"]
```

♦ **Q8. Given a String, how do you find the character with the highest frequency?**

🧑 **Logic:**

```
String s="bbaaac";

Map<Character, Long> map = s.chars()
    .mapToObj(c -> (char) c)
    .collect(Collectors.groupingBy(c -> c, Collectors.counting()));

long max = map.values().stream().mapToLong(v -> v).max().orElse(0);

map.entrySet().stream()
    .filter(e -> e.getValue() == max)
    .forEach(System.out::println);
```

✅ **Output:**

```
a=3
```

♦ **Q9. Given a sentence, how do you find duplicate words along with their occurrence count, sorted by frequency in descending order?**

 **Logic:**

```
String sentence="Java is Java and Java is great";

Arrays.stream(sentence.split(" "))
    .collect(Collectors.groupingBy(w -> w, Collectors.counting()))
    .entrySet().stream()
    .sorted(Map.Entry.<String,Long>comparingByValue().reversed())
    .forEach(System.out::println);
```

 **Output:**

```
Java=3
is=2
```

♦ **Q10.How do you print the top 3 longest Strings from a list using Java 8 Streams?**

 **Logic:**

```
List<String> list=Arrays.asList("apple","banana","cherry","watermelon","kiwi","strawberry");

list.stream()
    .sorted(Comparator.comparingInt(String::length).reversed())
    .limit(3)
    .forEach(System.out::println);
```

 **Output:**

```
watermelon
strawberry
banana
```