**Basic Level:**

1. **Filtering Even Numbers:**
   * Given a list of integers, find all even numbers using Stream API.​[blog.devgenius.io](https://blog.devgenius.io/java-8-coding-and-programming-interview-questions-and-answers-62512c44f062)
2. **Filtering Strings Starting with a Specific Character:**
   * Given a list of strings, find all strings that start with the character 'A' using Stream API.​
3. **Finding the Longest String in a List:**
   * Given a list of strings, find the longest string using Stream API.​
4. **Calculating the Average of Numbers:**
   * Given a list of integers, calculate the average using Stream API.​
5. **Mapping Strings to Their Lengths:**
   * Given a list of strings, create a list of their lengths using Stream API.​
6. **Removing Duplicates from a List:**
   * Given a list of integers with duplicates, remove duplicates using Stream API.​
7. **Sorting Strings Alphabetically:**
   * Given a list of strings, sort them alphabetically using Stream API.​
8. **Counting Occurrences of a Specific Element:**
   * Given a list of strings, count how many times the string "Java" appears using Stream API.​
9. **Converting a List to a Set:**
   * Given a list of integers, convert it to a set to remove duplicates using Stream API.​
10. **Checking if All Elements Are Positive:**
    * Given a list of integers, check if all elements are positive using Stream API.
11. **Finding the First Element:**
    * Given a list of integers, find the first element using Stream API.
12. **Finding the Maximum Value:**
    * Given a list of integers, find the maximum value using Stream API.
13. **Finding the Minimum Value:**
    * Given a list of integers, find the minimum value using Stream API.
14. **Concatenating Strings:**
    * Given a list of strings, concatenate them into a single string separated by commas using Stream API.
15. **Filtering Strings by Length:**
    * Given a list of strings, find all strings with a length greater than 5 using Stream API.
16. **Mapping Integers to Their Squares:**
    * Given a list of integers, create a list of their squares using Stream API.
17. **Filtering Strings Containing a Substring:**
    * Given a list of strings, find all strings that contain the substring "Java" using Stream API.
18. **Sorting Numbers in Descending Order:**
    * Given a list of integers, sort them in descending order using Stream API.
19. **Calculating the Product of Numbers:**
    * Given a list of integers, calculate the product of all numbers using Stream API.
20. **Finding Distinct Characters in a String:**
    * Given a string, find all distinct characters using Stream API.

**Intermediate Level:**

1. **Grouping Strings by Their First Character:**
   * Given a list of strings, group them by their first character using Stream API.​
2. **Partitioning Numbers into Even and Odd:**
   * Given a list of integers, partition them into even and odd numbers using Stream API.​
3. **Counting Occurrences of Each Element:**
   * Given a list of strings, count the occurrences of each string using Stream API.​
4. **Flattening a List of Lists:**
   * Given a list of lists of integers, flatten it into a single list using Stream API.​
5. **Finding the Second Highest Number:**
   * Given a list of integers, find the second-highest number using Stream API.​
6. **Calculating the Sum of Squares:**
   * Given a list of integers, calculate the sum of their squares using Stream API.​
7. **Removing Null Values from a List:**
   * Given a list of strings with some null values, remove all null values using Stream API.​
8. **Sorting Strings by Length:**
   * Given a list of strings, sort them by length using Stream API.​
9. **Finding the Longest String in a List:**
   * Given a list of strings, find the longest string using Stream API.​
10. **Finding Strings with Unique Characters:**
    * Given a list of strings, find all strings that have all unique characters using Stream API.
11. **Calculating the Average Length of Strings:**
    * Given a list of strings, calculate the average length of the strings using Stream API.
12. **Finding Strings That Are Palindromes:**
    * Given a list of strings, find all strings that are palindromes using Stream API.
13. **Grouping Numbers by Their Remainder When Divided by 3:**
    * Given a list of integers, group them by their remainder when divided by 3 using Stream API.
14. **Finding the First Non-Repeated Character in a String:**
    * Given a string, find the first non-repeated character using Stream API.
15. **Calculating the Frequency of Words in a List:**
    * Given a list of strings, calculate the frequency of each word using Stream API.
16. **Finding Strings That Are Anagrams of Each Other:**
    * Given a list of strings, find all groups of strings that are anagrams using Stream API.
17. **Finding the Median of a List of Numbers:**
    * Given a list of integers, find the median value using Stream API.

 **Implementing Parallel Streams for Performance Optimization:**

* Given a large dataset, implement parallel streams to optimize performance for operations like filtering and mapping.​
* **Reference:** Java Stream Hard Interview Questions

 **Creating a Custom Collector:**

* Design a custom collector that collects elements into a Map with keys as the element's first character and values as lists of strings starting with that character.​
* **Reference:** [Java 8 Stream API Interview Questions and Answers](https://medium.com/thefreshwrites/java-8-stream-api-interview-questions-and-answers-dd559ebffb35)

 **Performing Grouping and Counting:**

* Given a list of strings, group them by their length and count the number of strings in each group.​
* **Reference:** [Top 20 Java 8 Stream API Operations](https://www.youtube.com/watch?v=KGvmG6sxMds)

 **Implementing Sliding Window Operations:**

* Given a list of integers, implement a sliding window of size N and compute the sum of elements within each window.​
* **Reference:** Java Stream Hard Interview Questions

 **Handling Infinite Streams:**

* Generate an infinite stream of numbers starting from 1 and find the first 100 prime numbers using Stream API.​
* **Reference:** [Java 8 Coding and Programming Interview Questions and Answers](https://blog.devgenius.io/java-8-coding-and-programming-interview-questions-and-answers-62512c44f062)

 **Implementing Custom Reduction Operations:**

* Implement a custom reduction operation to concatenate strings in a list, separated by a comma.​
* **Reference:** Java Stream Hard Interview Questions

 **Performing Nested Mapping:**

* Given a list of lists of integers, flatten the list and find the sum of squares of all numbers.​
* **Reference:** [Java 8 Stream API Interview Questions and Answers](https://medium.com/thefreshwrites/java-8-stream-api-interview-questions-and-answers-dd559ebffb35)

 **Implementing Peek for Debugging:**

* Use the peek method to debug a stream pipeline by printing elements at various stages.​[cloudfoundation.com](https://cloudfoundation.com/blog/java-stream-interview-questions-and-answers/)
* **Reference:** Java Stream Hard Interview Questions

 **Performing Parallel Reduction:**

* Given a list of integers, calculate the sum using parallel streams and compare performance with sequential streams.​
* **Reference:** [Java 8 Stream API Interview Questions and Answers](https://medium.com/thefreshwrites/java-8-stream-api-interview-questions-and-answers-dd559ebffb35)

 **Implementing Custom Spliterators:**

* Create a custom spliterator to traverse a collection in a non-standard manner, such as in reverse order.
* **Reference:** Java Stream Hard Interview Questions

 **Handling Optional Values in Streams:**

* Given a list of strings, find the first string that starts with 'A' and convert it to uppercase, handling cases where no such string exists.
* **Reference:** [Java 8 Stream API Interview Questions and Answers](https://medium.com/thefreshwrites/java-8-stream-api-interview-questions-and-answers-dd559ebffb35)

 **Implementing Parallel Streams with Stateful Operations:**

* Demonstrate the challenges of using parallel streams with stateful operations like sorted and how to address them.
* **Reference:** Java Stream Hard Interview Questions

 **Creating a Stream from a File:**

* Read a large text file line by line into a stream and process each line to count word frequencies.
* **Reference:** [Java 8 Stream API Interview Questions and Answers](https://medium.com/thefreshwrites/java-8-stream-api-interview-questions-and-answers-dd559ebffb35)

 **Implementing a Stream Pipeline with Multiple Collectors:**

* Given a list of strings, create a pipeline that collects strings into a list and a set simultaneously.
* **Reference:** Java Stream Hard Interview Questions

 **Performing Parallel Stream Operations with Side Effects:**

* Discuss the implications of using side-effecting operations within parallel streams and how to mitigate issues.
* **Reference:** Java Stream Hard Interview Questions

 **Implementing a Stream Pipeline with Multiple Terminal Operations:**

* Given a list of integers, implement a pipeline that computes the sum and finds the maximum value in a single traversal.
* **Reference:** Java Stream Hard Interview Questions

 **Using Streams with Custom Data Structures:**

* Given a custom data structure, implement methods to convert it to a stream and perform various stream operations.
* **Reference:** [Java 8 Stream API Interview Questions and Answers](https://medium.com/thefreshwrites/java-8-stream-api-interview-questions-and-answers-dd559ebffb35)

 **Implementing Lazy Evaluation with Streams:**

* Demonstrate lazy evaluation by creating a stream pipeline where intermediate operations are not executed until a terminal operation is invoked.
* **Reference:** [Java Stream Hard Interview Questions](https://medium.com/@mehar.chand.cloud/java-stream-hard-interview-questions-54ea0