Lambda Expressions

1. **What are lambda expressions in Java 8?**
    - Explain what lambda expressions are and their syntax.
2. **What are the main benefits of using lambda expressions?**
    - Discuss the advantages lambda expressions bring to Java programming.
3. **Can you provide an example of a lambda expression?**
    - Write a simple code snippet using a lambda expression.
4. **What is the target type of a lambda expression?**
    - Explain what target type means in the context of lambda expressions.

Functional Interfaces

5. **What is a functional interface in Java 8?**
    - Define a functional interface and provide an example.
6. **Name some of the built-in functional interfaces provided in Java 8.**
    - List interfaces such as Predicate, Function, Consumer, and Supplier.
7. **What is the @FunctionalInterface annotation?**
    - Describe the purpose and use of the @FunctionalInterface annotation.

Streams API

8. **What is the Streams API in Java 8?**
    - Explain the Streams API and its purpose.
9. **What are the main differences between sequential and parallel streams?**
    - Discuss how sequential and parallel streams differ and when to use each.
10. **Can you demonstrate a simple use case of the Streams API?**
    - Provide a code example that uses the Streams API to filter and process data.
11. **What are intermediate and terminal operations in the context of streams?**
    - Define intermediate and terminal operations and provide examples of each.

Default and Static Methods in Interfaces

12. **What are default methods in interfaces?**
    - Explain what default methods are and why they were introduced.
13. **What are static methods in interfaces?**
    - Describe static methods in interfaces and how they differ from default methods.
14. **Can you provide an example of a default method in an interface?**

- Write a simple interface with a default method.

Optional Class

15. **What is the Optional class in Java 8?**
    - Explain the purpose of the Optional class and how it helps in avoiding null pointer exceptions.
16. **How do you create an Optional object?**
    - Provide examples of creating Optional objects using methods like of, ofNullable, and empty.
17. **How do you use the ifPresent method of the Optional class?**
    - Demonstrate the use of ifPresent with a code example.

Date and Time API

18. **What improvements were made to date and time handling in Java 8?**
    - Discuss the new java.time package and its advantages over the old java.util.Date and java.util.Calendar.
19. **What is the LocalDate class in Java 8?**
    - Explain the LocalDate class and provide an example of its usage.
20. **What is the LocalDateTime class and how is it different from LocalDate?**
    - Describe the LocalDateTime class and provide an example of its usage.

Other Enhancements

21. **What are the improvements to the java.util.concurrent package in Java 8?**
    - Mention enhancements like CompletableFuture and new methods in existing classes.
22. **What is the Collectors class and how is it used in the Streams API?**
    - Explain the role of the Collectors class and provide an example of collecting stream results.
23. **What is the purpose of the java.util.function package?**
    - Discuss the purpose of this package and some of the key interfaces it contains.
24. **What are method references in Java 8?**
    - Explain what method references are and provide examples of different types of method references.

Practical Questions

25. **Write a simple program using a stream to filter and collect a list of strings that start with a specific letter.**

- Provide a task to write a program that demonstrates the use of streams and lambda expressions.

26. **Demonstrate how to use the Optional class to avoid a null pointer exception in a method.**
    - Ask for a code example that uses Optional to handle potential null values.
27. **Write an example of using a lambda expression to sort a list of strings by their length.**
    - Provide a task to write a program that sorts a list using a lambda expression.