1. **Basic Concepts**
   * What is the difference between JDK, JRE, and JVM?
   * Explain the concept of Java Virtual Machine (JVM). How does it work?
   * What are the main features of Java?
2. **Data Types and Variables**
   * What are the different types of data types in Java?
   * Explain the difference between primitive and reference data types in Java.
3. **Operators and Control Statements**
   * Describe the different types of operators in Java.
   * What are the different types of control statements in Java?
4. **Classes and Objects**
   * What is a class and an object in Java?
   * Explain the concept of constructors in Java.
   * What is the difference between a constructor and a method?
5. **Inheritance and Polymorphism**
   * What is inheritance in Java?
   * Explain the concept of method overloading and method overriding.
   * What is polymorphism? Provide an example.
6. **Interfaces and Abstract Classes**
   * What is the difference between an interface and an abstract class?
   * Can a class implement multiple interfaces? Can an interface extend another interface?

**Advanced Java**

1. **Exception Handling**
   * What is exception handling in Java? Why is it important?
   * Explain the difference between checked and unchecked exceptions.
   * How do you create a custom exception in Java?
2. **Collections Framework**
   * What are the main interfaces of the Java Collections Framework?
   * Explain the difference between ArrayList and LinkedList.
   * What is the difference between HashMap and TreeMap?
3. **Multithreading and Concurrency**
   * What is the difference between Thread class and Runnable interface?
   * Explain the concept of synchronization in Java.
   * What are some common problems of multithreading?
4. **File I/O and Serialization**
   * How do you read and write files in Java?
   * What is serialization? How do you serialize and deserialize an object?

**Data Structures and Algorithms**

1. **Array and String**
   * How do you reverse an array in Java?
   * Write a Java program to check if a given string is a palindrome.
   * How do you remove duplicates from an array?
2. **Linked List**
   * How do you reverse a linked list in Java?
   * Write a program to detect a cycle in a linked list.
   * How do you merge two sorted linked lists?
3. **Stacks and Queues**
   * Implement a stack using arrays.
   * Implement a queue using linked lists.
   * Explain the concept of a priority queue.
4. **Trees and Graphs**
   * How do you traverse a binary tree (in-order, pre-order, post-order)?
   * Write a program to find the height of a binary tree.
   * How do you implement a graph in Java?
5. **Sorting and Searching**
   * Explain the difference between bubble sort, selection sort, and insertion sort.
   * Write a program to implement quicksort.
   * How do you perform a binary search on a sorted array?

**Design Patterns and Best Practices**

1. **Design Patterns**
   * What are design patterns? Why are they important?
   * Explain the Singleton design pattern. How do you implement it in Java?
   * What is the Factory design pattern?
2. **Best Practices**
   * What are some best practices for writing clean and efficient Java code?
   * How do you optimize the performance of a Java application?
   * Explain the concept of code refactoring.

**Miscellaneous**

1. **Java 8 Features**
   * What are the new features introduced in Java 8?
   * Explain the concept of lambda expressions.
   * What are streams in Java 8? How do you use them?
2. **Spring Framework**
   * What is Spring Framework? Why is it used?
   * Explain the concept of dependency injection in Spring.
   * What is Spring Boot? How does it simplify the development of Spring applications?
3. **Hibernate**
   * What is Hibernate? Why is it used?
   * Explain the concept of ORM (Object-Relational Mapping).
   * How do you configure and use Hibernate in a Java application?