Module 1:

1. Describe the evolution of Java from its inception to the present day. Include major versions and their contributions.
2. Explain the key features of Java that make it platform-independent and secure.
3. Discuss the importance of bytecode in the Java programming model and how it contributes to Java's platform independence.
4. What is the Java Virtual Machine (JVM)? Explain its architecture and role in executing Java applications.
5. Outline the components of the Java Development Kit (JDK) and explain their functions.
6. Analyze the structure of a simple Java program, explaining each part in detail.
7. Compare and contrast compiling and interpreting a Java program. Why Java is considered a compiled and interpreted language?
8. What are Java tokens? Discuss their significance and provide examples of each type.
9. List and explain the types of data types in Java. How do they differ in terms of usage and memory allocation?
10. Provide an overview of primitive and non-primitive data types in Java. Discuss when and why one would be used over the other.
11. Explain the different types of operators in Java. Provide examples for arithmetic, logical, and relational operators.
12. Describe how control statements are used in Java. Provide examples to demonstrate their practical applications.
13. What are the different looping structures in Java? Explain each one and give examples of when they would be used.

Module 2:

14. What is the difference between a class and an object in Java? Illustrate with examples.
15. Explain the process of creating an object in Java. What role does the `new` keyword play?
16. Define an object reference in Java and explain its significance in object-oriented programming.
17. Discuss the concept of method overloading in Java. Provide examples to illustrate how it works.
18. What is a constructor in Java? How does constructor overloading differ from method overloading? Include examples.
19. Describe how to pass and return objects from a method in Java. What are the implications of passing objects by reference?
20. Explain the purpose of the `new` operator in Java. How does it differ from the `new` keyword in other programming languages?
21. What are the roles of the `this` and `static` keywords in Java? Explain how they affect the behavior of classes and objects.
22. What is the `finalize()` method? When is it called, and what is its significance in garbage collection?
23. Discuss the different visibility modifiers in Java and their impact on class members.
24. What are nested classes and inner classes? Explain the different types and their use cases.

25. How does encapsulation benefit object-oriented programming? Provide examples of how it is implemented in Java.

Module 3:

26. Define inheritance in Java and explain its different types (single, multiple, multilevel, hierarchical, and hybrid). How does inheritance promote code reusability?
27. What is the member access rule in Java? Explain how it affects inheritance and access to class members.
28. Discuss the use of the `this` and `super` keywords in the context of inheritance. Provide examples demonstrating their usage.
29. What is an abstract class in Java? When should it be used, and how is it different from an interface?
30. Explain dynamic method dispatch in Java. How does it enable runtime polymorphism?
31. Discuss the use of the `final` keyword in Java and provide examples of where it can be applied (classes, methods, variables).
32. How do packages function in Java? Discuss the process of defining and importing packages.
33. What is an interface in Java? How is it different from an abstract class, and when would you choose one over the other?
34. Explain how to define and implement an interface in Java. What are the benefits of using interfaces?
35. How does Java handle multiple inheritance? Explain with examples of using interfaces to achieve this functionality.
36. What is polymorphism in Java? Describe its types and provide examples to illustrate both compile-time and runtime polymorphism.
37. Explain the role of the `extends` keyword in Java and how it is used to create subclasses.

Module 4:

38. What is a stream in Java? Explain the concept of byte streams and character streams, and provide examples of their usage.
39. How do byte streams differ from character streams? Discuss the advantages and disadvantages of each type.
40. Explain how to read input from the console in Java. What are the different ways to take user input, and when should each be used?
41. Discuss how to write output to the console in Java. Include examples of `System.out.print()` and `System.out.println()`.
42. What are exceptions in Java? Discuss the different types of exceptions and how they are managed.
43. Explain the usage of `try`, `catch`, `throw`, `throws`, and `finally` keywords in exception handling. Provide examples to show their practical use.
44. How do you create a custom exception class in Java? What is the advantage of creating your own exception?
45. What is a thread in Java? Explain the lifecycle of a thread and how to manage its different states.

46. How can a thread be created in Java using both the `Thread` class and the `Runnable` interface? Compare the two approaches.
47. What is thread priority, and how does it impact thread scheduling? Discuss how thread priorities can be set in Java.
48. Describe the AWT class hierarchy and how it is used to create graphical user interfaces.
49. What are the user interface components available in AWT? Explain their significance and usage with examples.
50. How are mouse and keyboard events handled in Java? Discuss the different event listener interfaces and their implementations.