Practical Questions For Learn Java Programming.

Module-1: Java Fundamentals

1. How do you download and install the latest version of Java on your system?
2. Explain the process of setting up the Java environment variables.
3. Walk me through the steps to download and install Eclipse IDE.
4. What are the key features and components of the Eclipse IDE?
5. Write a basic Java program that prints "Hello, World!" to the console.
6. How do you compile and run a Java program using the command line?
7. Discuss the differences between the `print` and `println` methods when used for console output.
8. Define a variable in Java and demonstrate different data types you can use.
9. Explain the concept of Java operators with suitable examples.
10. Write a Java program that takes an integer input and determines whether it's even or odd.
11. Create a loop that prints the numbers from 1 to 10, and then modify it to print only the odd numbers.
12. How would you use the `break` statement within a loop? Provide an example.
13. Implement a single-dimensional array that stores the names of three countries.
14. Create a 2D array to store a multiplication table from 1 to 5.
15. Manipulate a string using various String class methods like `length()`, `toUpperCase()`, and `substring()`.
16. Given a string, write a program to count the occurrences of a specific character.
17. Explain the difference between `equals()`, `equalsIgnoreCase()`, and `compareTo()` methods of the String class.

Module-2: Java OOPS Concepts

1. Define a class called "Car" with relevant attributes and methods.
2. Create an object of the "Car" class and call its methods.
3. How do you pass parameters to methods in Java? Provide examples.
4. Distinguish between call by value and call by reference with appropriate code snippets.
5. Write a class named "Student" with a parameterized constructor to initialize name and age.

6. Implement method overloading in the "Student" class by creating multiple constructors.
7. Create a class "MathUtils" with overloaded methods to add integers and doubles.
8. How does the `this` keyword work in Java? Illustrate its usage.
9. Develop a class "Circle" with a static variable for the value of Pi and a static method to calculate the area.
10. Create a superclass "Animal" and a subclass "Dog" demonstrating method overriding.
11. Explain the usage of the `super` keyword with an example in the context of method overriding.
12. How and why would you use the `final` keyword in Java? Provide scenarios.
13. Define an interface "Shape" with a method to calculate area. Implement it in classes like "Circle" and "Rectangle."
14. Organize classes into packages and demonstrate their usage.
15. Discuss the access modifiers (`private`, `protected`, `default`, and `public`) with appropriate examples.
16. Write a program that showcases exception handling using `try`, `catch`, and `finally` blocks.
17. Utilize an ArrayList to manage a list of names and perform basic operations like insertion and deletion.
18. Implement a HashMap to store and retrieve key-value pairs representing words and their meanings.
19. Outline the steps involved in establishing a JDBC connection to a database.
20. How would you retrieve data from a database using JDBC and display it in your Java program?
21.