**1. What is Programming?**  
Programming is the process of writing instructions for a computer to perform specific tasks. These instructions are written in a programming language. It helps in building software applications like websites, mobile apps, games, or operating systems. Programmers write algorithms to solve problems, then translate them into code the computer can understand.

**2. Why Do We Need Programming Languages?**  
Computers only understand binary (0s and 1s). Programming languages help us give commands in a readable form. These languages act as a bridge between humans and machines. They help write complex software, automate tasks, and solve problems efficiently.

**3. Why Java? What Makes It Special?**  
Java is platform-independent, meaning you can write code once and run it anywhere. It’s object-oriented, secure, and widely used in real-world applications like banking systems, Android apps, web apps, etc. Its large community, libraries, and support make it reliable for long-term projects.

**4. What is the Difference Between High-Level and Low-Level Languages?**  
Low-level languages (machine and assembly) are hardware-specific and hard to understand. High-level languages (Java, Python, etc.) are easy to write and understand. They are closer to human language and use compilers or interpreters to convert to machine code.

**5. Explain the Java Compilation and Execution Process.**  
Java source code (.java) is compiled by the **javac** compiler into **bytecode** (.class). This bytecode is platform-independent. The **JVM** (Java Virtual Machine) then interprets this bytecode into machine code suitable for your OS.  
Steps:

1. Write code → 2. Compile → 3. Bytecode → 4. JVM executes → 5. Output

**6. What is the Difference Between JDK, JRE, and JVM?**

* **JDK (Java Development Kit)**: Includes tools for development (compiler, debugger, etc.)
* **JRE (Java Runtime Environment)**: Only required to run Java programs (includes JVM + libraries)
* **JVM (Java Virtual Machine)**: Converts bytecode to machine code during execution.

**7. What is Bytecode?**  
Bytecode is an intermediate code generated after compilation of a Java file. It is not human-readable but platform-independent. The .class file contains this bytecode. The JVM uses it to run the program on any OS.

**8. What are the Steps to Install Java on a System?**

1. Download JDK from [Oracle](https://www.oracle.com/java/technologies/javase-downloads.html)
2. Install and configure environment variable JAVA\_HOME
3. Add JDK's bin folder to the PATH variable
4. Test using java -version and javac commands in CMD/Terminal

**9. What are Different Programming Languages?**  
There are low-level (Assembly, Machine) and high-level (Java, Python, C++, etc.) languages.

* **Java** – secure, object-oriented
* **Python** – simple syntax
* **C/C++** – fast, system-level
* **JavaScript** – web scripting  
  Each language has its purpose depending on the field of application.

**10. How Do Computers Understand Programming Languages?**  
Computers can only process binary code. So, when we write code in Java or any other high-level language, it gets **compiled** or **interpreted** into machine code using tools like compilers (javac) or JVM.

**11. What is an Algorithm?**  
An algorithm is a step-by-step solution to a problem. It is a logical procedure used to solve tasks in a finite number of steps. Before writing code, a programmer often writes the algorithm to understand the problem clearly.

**12. Explain the .java and .class Files.**

* .java file: The source code written by the programmer
* .class file: The compiled bytecode generated by the javac compiler  
  The .class file is what the JVM uses to run the program.

**13. What is Machine Language?**  
Machine language is the lowest-level programming language, consisting only of binary (0s and 1s). It is directly understood by the CPU. Every high-level language code eventually gets converted to machine code for execution.

**14. What Do You Mean by Platform Independence in Java?**  
Java programs can run on any OS that has a JVM. Once compiled into bytecode, the .class file doesn’t depend on OS. This makes Java **“write once, run anywhere.”**

**15. What is the Role of the Java Virtual Machine (JVM)?**  
JVM loads .class files, verifies bytecode, and converts it to native machine code for the OS. It provides runtime features like garbage collection and security. JVM makes Java platform-independent.

**16. Why is Java Called Object-Oriented?**  
Java follows Object-Oriented Programming (OOP) principles. Everything in Java is treated as an object. It supports concepts like classes, objects, inheritance, encapsulation, polymorphism, and abstraction.

**17. Difference Between Compiler and Interpreter?**

* **Compiler**: Translates entire code at once into bytecode (Java uses javac)
* **Interpreter**: Executes code line-by-line (JVM acts as interpreter during execution)  
  Java uses both: Compiler to generate bytecode, and JVM to interpret/run it.

**18. What is the Use of the javac and java Commands?**

* javac MyFile.java → compiles source code to bytecode (.class)
* java MyFile → runs the compiled class using JVM  
  These are the two essential tools for Java developers.

**19. What is the Real-World Importance of Java?**  
Java is used in Android apps, web servers, banking apps, enterprise systems, and cloud computing. Companies like Google, Amazon, and Netflix rely on Java. Its portability, security, and performance make it ideal for long-term projects.

**20. What are the Components of Java Development Environment?**

* **Editor** (Notepad, VS Code, IntelliJ)
* **Compiler** (javac)
* **Interpreter/Runtime** (JVM)
* **Debugger**
* **JDK libraries**  
  Together, they provide tools for writing, compiling, testing, and running Java applications.