Q: What is Java?

A: Java is a high-level, class-based, object-oriented programming language developed by Sun Microsystems (now Oracle). It is platform-independent (WORA – Write Once, Run Anywhere) because of the JVM.

Q: Why Java?

A: Java is platform-independent, secure, robust, has rich APIs, strong community support, supports OOP principles, and has automatic memory management.

Q: Difference between Java and Python?

A: Java: Statically typed, faster, verbose syntax, used for enterprise apps & Android. Python: Dynamically typed, slower, simple syntax, widely used in AI/ML & scripting.

Q: Do you think Java is 100% object-oriented?

A: No. Java is not 100% object-oriented because it supports primitive data types (`int`, `float`). Wrapper classes (`Integer`, `Float`) allow primitives to behave like objects.

Q: What is Object-Oriented?

A: OOP is a paradigm based on objects containing data (fields) and methods (behavior), promoting reusability, abstraction, encapsulation, and modularity.

Q: Features of Java

A: Simple, Secure, Platform Independent, Robust, Portable, Object-Oriented, Multithreaded, High Performance (with JIT), Automatic Garbage Collection.

Q: What is DataType?

A: Data type defines the type of data a variable can hold, e.g., int, double, boolean, char.

Q: Difference between float and double

A: float: 32-bit, single precision (~7 digits). double: 64-bit, double precision (~15 digits).

Q: What is Access Modifiers?

A: Keywords that set scope/visibility of classes, methods, and variables: public, private, protected, default.

Q: Difference between private and protected

A: private: Accessible only within the same class. protected: Accessible within the same package + subclasses.

Q: What is a Variable?

A: A container that stores data values, e.g., int age = 25;

Q: Java Syntax Example

A: class Demo { public static void main(String[] args) { System.out.println("Hello, Java!"); } }

Q: What is Static?

A: The static keyword means the member belongs to the class, not instances. Example: static int count;

Q: Difference between JDK, JRE, JVM, JIT

A: JDK: Development Kit, JRE: Runtime Environment, JVM: Executes bytecode, JIT: Optimizes runtime performance.

Q: What are 2 pillars of Java?

A: JVM and OOP (Object-Oriented Programming).

Q: What are 4 pillars of OOP?

A: Encapsulation, Abstraction, Inheritance, Polymorphism.

Q: Explain 4 pillars of OOP

A: Encapsulation: Wrapping data/methods. Abstraction: Hiding implementation. Inheritance: Parent-child relation. Polymorphism: Many forms (overloading/overriding).

Q: What is 'this' keyword?

A: Refers to the current object. Used to access instance variables, methods, or constructors.

Q: What is 'super' keyword?

A: Refers to parent class object. Used to access parent methods/variables and constructors.

Q: Difference between ArrayList and LinkedList

A: ArrayList: Dynamic array, fast access, slower insertion. LinkedList: Doubly linked, slower access, faster insertion/deletion.

Q: Difference between if-else and switch

A: if-else: Works for complex conditions. switch: Works for discrete values like int, char, enum, String.

Q: Types of Loops in Java

A: for, while, do-while, for-each.

Q: Difference between while and do-while

A: while: Condition first, may not execute. do-while: Executes at least once.

Q: Difference between for-each and for

A: for: Index-based, more control. for-each: Iterates directly over elements.

Q: Difference between import java.util.* and import java.util.ArrayList

A: import java.util.*: Imports all classes. import java.util.ArrayList: Imports only ArrayList.

Q: Difference between public static void main and main

A: public static void main: JVM entry point. main: Won’t work as entry unless explicitly called.

Q: What is Memory Management in Java?

A: Managed by JVM with Heap (objects) & Stack (methods, vars). Garbage Collector clears unused objects.

Q: Can we use break outside loop/switch?

A: No. break can only be used inside a loop or switch-case. Outside causes compile-time error.