1. What is Java?

- Java is a high-level, object-oriented programming language developed by Sun Microsystems in 1995. It's designed to have minimal implementation dependencies, making it platform- independent.

2. What is the JDK?

- The JDK (Java Development Kit) is a software development environment used for developing Java applications. It includes the Java Runtime Environment (JRE), a compiler, and other development tools.

3. What is the JRE?

- The JRE (Java Runtime Environment) is the runtime portion of Java software, which is all you need to run Java applications but not to develop them

4. What is the JVM?

- The JVM (Java Virtual Machine) is a platform-independent execution environment that converts Java bytecode into machine language and executes it.

5. What is bytecode?

Bytecode is the intermediate code generated after the Java compiler compiles the source code. It is platform-independent and can be executed by the JVM.

6. What is the difference between JDK, JRE, and JVM?

- JDK: For development (contains JRE + tools for development).
- JRE: For running Java applications (contains JVM + libraries).
- JVM: For executing bytecode (runs Java programs).

7. Is Java platform-independent? Why?

Yes, Java is platform-independent because of its bytecode that can run on any system that has a JVM, regardless of the underlying architecture.

8. What is the main feature of the Java language?

The primary feature of Java is platform independence, achieved through the use of the JVM.

9. What is object-oriented programming (OOP)?

- OOP is a programming paradigm that relies on the concept of classes and objects. Key principles of OOP include encapsulation, inheritance, polymorphism, and abstraction.

10. What is a class in Java?

- A class in Java is a blueprint from which objects are created. It can contain fields (attributes) and methods (functions) to define the behavior and state of objects.

11. What is an object in Java?

- An object is an instance of a class. It is a real-world entity that holds state and behavior defined by its class

12. What is encapsulation?

- Encapsulation is the wrapping of data (fields) and code (methods) into a single unit, known as a class, and restricting direct access to some components.

13. What is inheritance?

- Inheritance is a mechanism where one class acquires the properties (fields and methods) of another class. It helps in code reusability.

14. What is polymorphism?

Polymorphism is the ability of a method to perform different functions based on the object that it is acting upon. It allows one interface to be used for a general class of actions.

15. What is abstraction?

Abstraction is the concept of hiding the internal implementation of a feature and only exposing the functionality to the user.

16. What is the difference between an interface and an abstract class?

- Interface: Can only have abstract methods and final fields (Java 8+ allows default and static methods).
- Abstract class: Can have both abstract and concrete methods and can hold state.

17. What is a constructor in Java?

A constructor is a special method used to initialize objects. It is invoked when an object of a class is created.

18. What is the default constructor?

A default constructor is a constructor provided by Java if no other constructors are defined in the class. It has no parameters.

19. What is method overloading?

- Method overloading is when a class has more than one method with the same name but different parameter lists (either in number or type).

20. What is method overriding?

- Method overriding occurs when a subclass provides a specific implementation of a method that is already defined in its superclass.

21. What is the 'super' keyword in Java?

- The super keyword refers to the immediate parent class object. It is used to access parent class methods or constructors.

22. What is a package in Java?

A package is a namespace that organizes classes and interfaces. It helps to avoid name conflicts and to control access.

23. What is the use of the 'final' keyword?

- The final keyword can be used with classes, methods, and variables to indicate that they cannot be modified or extended.

24. What is the static keyword?

The static keyword in Java is used for memory management. It can be applied to variables, methods, blocks, and nested classes. Static members belong to the class rather than instances.

25. What is the difference between static and instance variables?

- Static variables are shared among all instances of a class.
- Instance variables are unique to each instance of a class.

26. What is the 'this' keyword in Java?

The This keyword is used to refer to the current instance of the class.

27. What is exception handling?

Exception handling in Java is a mechanism to handle runtime errors, allowing normal flow of the application even after an error occurs.

28. What is the difference between checked and unchecked exceptions?

- Checked exceptions are checked at compile-time.
- Unchecked exceptions are checked at runtime (subclasses of RuntimeException).

29. What is the try-catch block?

- The try block contains code that might throw an exception, while the catch block handles the exception if it occurs.

30. What is the finally block in Java?

- The finally block contains code that is always executed, regardless of whether an exception is thrown or caught.

31. What is the throw keyword in Java?

- The throw keyword is used to explicitly throw an exception.

32. What is the throws keyword?

- The throws keyword is used in method signatures to declare that a method may throw exceptions.

33. What are collections in Java?

- The Java Collections Framework provides a set of interfaces and classes to store and manipulate data in groups.

34. What is the difference between an Array and an ArrayList?

- • Arrays are of fixed size, while ArrayLists are resizable.
  • Arrays can hold primitive types, whereas ArrayLists hold objects.