

7. What is method overriding?Method overriding occurs when a subclass provides a specific implementation of a method already defined in the parent class. The method name, return type, and parameters must be the same. It is an example of runtime polymorphism. Overriding allows subclasses to change or extend the behavior of inherited methods. The @Override annotation is commonly used. It helps in achieving dynamic method dispatch and improves flexibility and extensibility in Java applications.

8. Define inheritance in Java.Inheritance is an Object-Oriented Programming feature where one class acquires properties and methods of another class. The existing class is called the parent or superclass, and the new class is called the child or subclass. Inheritance promotes code reusability and reduces duplication. Java supports single, multilevel, and hierarchical inheritance using the extends keyword. It also helps in achieving polymorphism and establishing relationships between classes in Java applications.

9. What are access modifiers?Access modifiers in Java are keywords used to control the visibility and accessibility of classes, variables, methods, and constructors. Java provides four access modifiers: public, private, protected, and default. Public members are accessible everywhere, private members only within the same class, protected members within the package and subclasses, and default members within the same package only. Access modifiers improve security, data hiding, and proper access control in Java programs.

10. What is the use of this keyword?The this keyword in Java refers to the current object of a class. It is used to distinguish instance variables from local variables when they have the same name. It can also be used to call constructors, methods, and pass the current object as an argument. The this keyword improves code clarity and avoids confusion. It is mainly used in constructors and methods to access current class members.

11. What is static member in Java?A static member in Java belongs to the class rather than objects. Static variables and methods are shared among all objects of the class. They can be accessed using the class name without creating an object. Static members save memory because only one copy exists. Static methods can directly access only static data members. The main() method in Java is static because it must run before any object creation.

12. What is garbage collection?Garbage collection in Java is an automatic memory management process that removes unused objects from memory. It helps in freeing heap memory and prevents memory leaks. The Java Virtual Machine (JVM) automatically performs garbage collection when objects are no longer referenced. Programmers do not need to manually delete objects. The System.gc() method can request garbage collection, but execution is controlled by the JVM. Garbage collection improves memory efficiency and application performance

7. Define throw keyword.The throw keyword in Java is used to explicitly create and throw an exception. It allows programmers to generate custom exceptions manually according to program requirements. The throw statement is followed by an exception object. For example, throw new ArithmeticException("Invalid operation"); creates and throws an exception. When an exception is thrown, control transfers to the nearest matching catch block. The throw keyword helps in implementing customized and controlled exception handling in Java programs.

8. Define throws keyword.The throws keyword in Java is used in method declarations to specify the exceptions that a method may generate. It informs the caller of the method about possible exceptions so that they can be handled properly. Multiple exceptions can be declared using throws. It is mainly used with checked exceptions. For example, public void readFile() throws IOException specifies that the method may throw an IOException during execution. It improves program safety and reliability.

9. What are checked exceptions?Checked exceptions are exceptions checked by the Java compiler during compilation. These exceptions must be handled using try-catch blocks or declared using the throws keyword. They occur due to external conditions such as file handling or database access. Examples include IOException and SQLException. Checked exceptions improve program reliability by forcing programmers to handle possible errors. If checked exceptions are not handled properly, the program will not compile successfully in Java.

10. What are unchecked exceptions?Unchecked exceptions are exceptions that are not checked by the compiler during compilation. They occur mainly because of programming mistakes such as dividing by zero or accessing invalid array indexes. These exceptions belong to the RuntimeException class. Examples include ArithmeticException, NullPointerException, and ArrayIndexOutOfBoundsException. Unchecked exceptions are not compulsory to handle using try-catch blocks. They indicate logical errors in the program and usually occur during runtime execution in Java applications.

11. What is multiple catch block?A multiple catch block in Java means using more than one catch block with a single try block to handle different types of exceptions separately. Each catch block handles a specific exception type. When an exception occurs, Java checks the catch blocks in order and executes the matching one. Multiple catch blocks improve error handling and make programs more organized. They allow programmers to display different messages and take suitable actions for different exceptions.

12. What is nested try statement?A nested try statement in Java means placing one try block inside another try block. It is used when different parts of a program require separate exception handling mechanisms. If an exception occurs in the inner try block, it is first handled there. If not handled, control passes to the outer catch block. Nested try statements improve flexibility and allow better management of complex programs where multiple exceptions may occur at different levels.

7. What are Java buzzwords?Java buzzwords are special features that describe the characteristics of Java. Important buzzwords include simple, object-oriented, secure, portable, platform-independent, robust, multithreaded, distributed, dynamic, and high performance. These features make Java suitable for modern software development. For example, platform independence allows Java programs to run on any system, while security protects applications from harmful operations. Java buzzwords explain why Java became one of the most popular programming languages worldwide.

8. Define variables in Java.Variables in Java are named memory locations used to store data values during program execution. Each variable has a specific data type that defines the kind of value it can hold, such as integer, float, or character. Variables can change their values during program execution. In Java, variables must be declared before use. Examples include int age = 20; and float salary = 5000.5f. Variables help in performing calculations and storing program data efficiently.

9. What are data types in Java?Data types in Java define the type and size of data that variables can store. Java data types are divided into two categories: primitive and non-primitive. Primitive data types include int, float, char, double, boolean, byte, short, and long. Non-primitive data types include arrays, classes, and strings. Data types help the compiler allocate memory efficiently and prevent invalid operations. Choosing proper data types improves program performance, accuracy, and memory management in Java applications.

10. Explain decision-making statements in Java.Decision-making statements in Java are used to control program flow based on conditions. They allow the program to choose different actions depending on true or false results. Common decision-making statements are if, if-else, nested if, else-if ladder, and switch statement. For example, if marks are greater than 40, the program displays "Pass." These statements help implement logical conditions in programs and make software intelligent, flexible, and capable of handling different situations effectively.

11. What are looping statements in Java?Looping statements in Java are used to execute a block of code repeatedly until a condition becomes false. Java provides three main loops: for loop, while loop, and do-while loop. The for loop is used when the number of iterations is known, while while and do-while loops are used for condition-based repetition. Loops reduce code duplication, save time, and improve program efficiency. They are commonly used in arrays, calculations, and repetitive tasks.

12. What is a method in Java?A method in Java is a block of code used to perform a specific task. Methods improve code reusability and reduce repetition. They are defined inside a class and are called when needed. A method may accept parameters and may return a value. Examples include display(), add(), and calculate(). Methods make programs modular and easier to maintain. Java provides predefined methods and also allows programmers to create user-defined methods for customized operations.

7. What is thread synchronization?Thread synchronization is a mechanism used to control access to shared resources by multiple threads. It prevents data inconsistency and thread interference during concurrent execution. In Java, synchronization is achieved using the synchronized keyword. Only one thread can access synchronized code or methods at a time. Synchronization is important when multiple threads modify shared data. It ensures data integrity, thread safety, and reliable execution of multithreaded Java applications without conflicts or errors.

8. What is inter-thread communication?Inter-thread communication is a mechanism that allows threads to communicate and coordinate with each other during execution. In Java, it is achieved using methods like wait(), notify(), and notifyAll(). These methods belong to the Object class. Inter-thread communication helps avoid busy waiting and improves efficient resource sharing between threads. It is mainly used in producer-consumer problems and synchronized applications where one thread waits for another thread to complete a specific task.

9. What is the use of sleep() method?The sleep() method in Java temporarily pauses the execution of a thread for a specified period of time. It belongs to the Thread class. During sleep, the thread remains inactive but does not release locks it holds. The method helps control thread execution speed and synchronization. For example, Thread.sleep(1000); pauses execution for one second. It is commonly used in animations, timers, and multithreading applications to create delays between operations.

10. What is the use of join() method?The join() method in Java is used to make one thread wait until another thread completes its execution. It belongs to the Thread class. When a thread calls join() on another thread, the current thread pauses until the target thread finishes. This method helps coordinate thread execution and maintain proper sequence in multithreaded programs. It is useful when the result of one thread is required before continuing execution of another thread.

11. Difference between extending Thread and implementing Runnable.In Java, a thread can be created either by extending the Thread class or implementing the Runnable interface. Extending Thread directly creates a thread class, but it prevents multiple inheritance because Java does not support multiple class inheritance. Implementing Runnable separates task logic from thread execution and allows the class to inherit another class. Runnable is preferred because it improves flexibility, code sharing, and better object-oriented design in multithreaded Java applications.

12. What is deadlock?Deadlock is a situation in multithreading where two or more threads wait indefinitely for resources locked by each other. As a result, none of the threads can continue execution. Deadlock usually occurs because of improper synchronization and resource sharing. It reduces application performance and may freeze the program completely. Deadlock can be avoided by proper lock management, avoiding circular waiting, and acquiring resources in a fixed order during thread synchronization in Java programs.

7. What is package in Java?A package in Java is a collection of related classes, interfaces, and sub-packages organized together. Packages help in avoiding naming conflicts and provide access protection. They improve code organization and reusability. Java provides predefined packages such as java.lang and java.util, and programmers can also create user-defined packages. Packages are declared using the package keyword. They make large programs easier to manage and help in developing modular and maintainable Java applications efficiently.

8. What is the use of super keyword?The super keyword in Java refers to the immediate parent class object. It is used to access parent class variables, methods, and constructors. The super keyword helps resolve ambiguity when parent and child classes have members with the same name. It is commonly used in inheritance. For example, super() calls the parent class constructor, and super.display() calls the parent class method. It improves code clarity and supports proper inheritance implementation in Java.

9. What is method overriding?Method overriding occurs when a subclass provides a new implementation of a method already defined in its parent class. The method name, parameters, and return type must remain the same. It is an example of runtime polymorphism in Java. Method overriding allows subclasses to modify inherited behavior according to their requirements. The @Override annotation is commonly used for clarity. It improves flexibility, extensibility, and dynamic behavior in Object-Oriented Programming applications.

10. What is nested interface?A nested interface is an interface declared inside another class or interface. It helps logically group related interfaces together and improves code organization. Nested interfaces can be public, private, or protected depending on their declaration. They are accessed using the outer class or interface name. Nested interfaces are useful when an interface is relevant only within another class or interface. They enhance encapsulation and make Java programs more structured and maintainable.

11. What is CLASSPATH?CLASSPATH is an environment variable in Java that tells the Java compiler and JVM where to find class files and packages. It helps Java locate user-defined classes, libraries, and external packages during program execution. If CLASSPATH is not set properly, Java may show "ClassNotFoundException" errors. The current directory is included by default. CLASSPATH can be set temporarily using commands or permanently through system environment settings for efficient Java application execution and management.

12. How do you import packages in Java?Packages in Java are imported using the import keyword. Importing packages allows programmers to use predefined classes and interfaces without writing full package names repeatedly. For example, import java.util.Scanner; imports the Scanner class, while import java.util.*; imports all classes of the java.util package. Packages are usually imported at the beginning of a Java program. Import statements improve readability, simplify coding, and provide access to useful predefined Java libraries and tools.

7. What is Set interface?The Set interface in Java is a part of the Collection Framework used to store unique elements only. It does not allow duplicate values. Sets are mainly used when uniqueness of data is important. Common implementations of the Set interface include HashSet, LinkedHashSet, and TreeSet. The Set interface provides methods for insertion, deletion, and searching operations. It improves efficient data handling and prevents duplication in Java applications and database-related tasks.

8. What is Queue interface?The Queue interface in Java is part of the Collection Framework and is used to store elements in a specific order for processing. It generally follows the FIFO (First In First Out) principle. Elements are inserted at the rear and removed from the front. Common classes implementing Queue are PriorityQueue and LinkedList. The Queue interface is widely used in scheduling, task management, and multithreading applications where ordered processing of elements is required.

9. Difference between ArrayList and LinkedList.ArrayList stores elements using a dynamic array, while LinkedList stores elements using doubly linked nodes. ArrayList provides faster random access because elements are indexed, whereas LinkedList provides faster insertion and deletion operations. ArrayList consumes less memory compared to LinkedList. LinkedList is better for frequent modifications, while ArrayList is suitable for searching and retrieval operations. Both classes implement the List interface and are widely used in Java Collection Framework applications.

10. What is HashSet?HashSet is a class in Java that implements the Set interface and stores unique elements only. It uses a hash table for data storage, which provides fast insertion, deletion, and searching operations. HashSet does not maintain insertion order and does not allow duplicate values. It can store null values. HashSet is useful when uniqueness and fast performance are required. It is commonly used in applications involving data filtering and duplicate removal operations.

11. What is JDBC?JDBC (Java Database Connectivity) is an API used to connect Java applications with databases. It provides methods for executing SQL queries, retrieving data, and updating database records. JDBC allows Java programs to interact with databases like MySQL, Oracle, and PostgreSQL. It includes classes and interfaces such as Connection, Statement, and ResultSet. JDBC makes database operations simple and platform independent, helping developers create database-driven applications efficiently in Java programming.

12. Explain steps to connect Java with database.To connect Java with a database using JDBC, first import JDBC packages. Then load the database driver using Class.forName(). After that, establish a connection using DriverManager.getConnection(). Create a Statement or PreparedStatement object to execute SQL queries. Execute queries using executeQuery() or executeUpdate(). Retrieve results using ResultSet if needed. Finally, close the connection and resources properly. These steps allow Java applications to interact efficiently with databases for data storage and management.