Java Technical Interview Questions

Core Java Concepts

1. What is the difference between JDK and JRE?
2. Why is Java a platform independent language?
3. What is the difference between an abstract class and an interface?
4. What is the difference between final, finally, and finalize?
5. What is the difference between stack and heap memory?
6. What is the difference between method overloading and method overriding?
7. What is the difference between an abstract class and an interface?
8. What is the difference between a private and a protected modifier?
9. What is constructor overloading in Java?
10. What is the use of super keyword in Java?
11. What is the difference between static methods, static variables, and static classes in Java?
12. What exactly is System.out.println in Java?
13. What part of memory - Stack or Heap - is cleaned in the garbage collection process?

Object-Oriented Programming

1. What are the Object Oriented Features supported by Java?
2. What are the different access specifiers used in Java?
3. What is the difference between composition and inheritance?
4. What is the purpose of an abstract class?
5. What are the differences between constructor and method of a class in Java?
6. What is the diamond problem in Java and how is it solved?
7. What is the difference between local and instance variables in Java?
8. What is a Marker interface in Java?

Data Structures and Algorithms

1. Why are strings immutable in Java?
2. What is the difference between creating a String using new() and as a literal?
3. What is the Collections framework?

4. What is the difference between ArrayList and LinkedList?
5. What is the difference between a HashMap and a TreeMap?
6. What is the difference between a HashSet and a TreeSet?
7. What is the difference between an Iterator and a ListIterator?
8. What is the difference between an ArrayList and a LinkedList?
9. What is the purpose of the Comparable interface?
10. What is the difference between a HashSet and a TreeSet?
11. What is the purpose of the java.util.concurrent package?

Exception Handling

1. What is an exception?
2. How does an exception propagate throughout the Java code?
3. What is the difference between checked and unchecked exceptions?
4. What is the use of try-catch block in Java?
5. What is the difference between throw and throws?
6. What is the use of the finally block?
7. What's the base class of all exception classes?
8. What is Java Enterprise Edition (Java EE)?
9. What is the difference between a Servlet and a JSP?
10. What is the purpose of the Java Persistence API (JPA)?
11. What is the difference between stateful and stateless session beans?

Multithreading

1. What is a thread and what are the different stages in its lifecycle?
2. What is the difference between process and thread?
3. What are the different types of thread priorities available in Java?
4. What is context switching in Java?
5. What is the difference between user threads and Daemon threads?
6. What is synchronization?
7. What is a deadlock?
8. What is the use of the wait() and notify() methods?
9. What is the difference between a thread and a process in Java?
10. What is the difference between synchronized and volatile in Java?
11. What is the purpose of the sleep() method in Java?
12. What is the difference between wait() and sleep() in Java?

13. What is the difference between notify() and notifyAll() in Java?

I finished my Technical Interview Questions asked :

1. Tell me about yourself.
2. Which language you are good at. (JAVA)
3. What is Object ?
4. What is Class ?
5. What is Constructor ?
6. Uses of Constructor ?
7. What is Object Orientation?
8. What is Abstraction?
9. What is Exception Handling ?
10. Any questions from your side ?

Interview Ended

Core Java Concepts Answers

Ans.1

JDK stands for Java Development Kit, which is a software development environment for building Java applications. JRE stands for Java Runtime Environment, which is required to run Java programs.

Ans.2

By relying on a virtual machine, Java achieves platform independence. In practice, this means that both the Java programming language and its associated APIs are first compiled into bytecodes that can run on multiple platforms. Then, the virtual machine handles any variations in how these bytecodes are executed across different platforms.

Ans.3

An abstract class is a class that cannot be instantiated and can only be inherited. An interface is a blueprint of a class that contains only abstract methods and constants.

Ans.4

Final is used to make a variable or method constant and cannot be changed later. finally is used in try-catch blocks to execute a block of code regardless of whether an exception is

thrown or not. finalise is a method that is called by the garbage collector when an object is no longer in use.

Ans.5

Stack memory is used for storing local variables and function call, while heap memory is used for storing objects and their instance variables.

Ans.6

Method overloading is creating multiple methods in a class with the same name but different parameters, while method overriding is creating a method in a subclass with the same name and parameters as a method in its superclass.

Ans.7

An abstract class can have both abstract and concrete methods, while an interface can only have abstract methods. A class can extend only one abstract class, but it can implement multiple interfaces.

Ans.8

A private modifier makes a member accessible only within the same class, while a protected modifier makes a member accessible within the same class and its subclasses.

Ans.9

Constructor overloading is a concept in object oriented programming where a class can have multiple constructors with different parameter lists. Each constructor provides a different way to initialise objects of that class.

Ans.10

The super keyword is used to access data members of the parent class when the data members names of the parent class and its child subclasses are the same, to call the default and parameterized constructor of the parent class inside the child subclass and to access parent class methods when the child subclasses have overridden them.

Ans.11

Static Methods and Static variables are those methods and variables that belong to the class of the java program, not to the object of the class. They are allocated memory when the class is loaded and can directly be called with the help of the class names. A class in the java program cannot be static except if it is the inner class. If it is an inner static class, then it exactly works like other static members of the class.

Ans.12

System.out.println() is a method to print a message on the console. System - It is a class present in java.lang package. Out is the static variable of type PrintStream class present in the System class. println() is the method present in the PrintStream class.

Ans.13

Garbage Collection is done on heap memory to free the memory used by objects that don't have any reference. Any object created in the heap space has global access and can be referenced from anywhere in the application.

Object-Oriented Programming

Ans.1

Java is an object-oriented programming language and supports the following object oriented features

- **Encapsulation:** Java allows encapsulation, which is the practice of hiding the implementation details of an object from other objects. This is achieved through the use of access modifiers
- **Inheritance:** Java supports inheritance, which allows a new class to be based on an existing class, inheriting its attributes and methods. This enables code reuse and makes it easier to create new classes that have common properties with existing classes.
- **Polymorphism:** Java supports polymorphism, which allows objects of different classes to be treated as if they were objects of a common superclass. This can be achieved through method overriding and method overloading
- **Abstraction:** Java allows abstraction, which is the process of hiding complex implementation details and providing a simplified interface for the user. This can be achieved through abstract classes and interfaces
- **Classes and Objects:** Java is a class-based language, which means that it provides constructs for defining classes and creating objects from those classes.

Ans.2

Java has 4 access specifiers.

- Public Can be accessed by any class or method
- Protected Can be accessed by the class of the same package, or by the sub-class of this class, or within the same class

- Default Are accessible only within the package, is the default option for all classes, methods and variables.
- Private Can be accessed only within the class

Ans.3

Composition is a "has-a" relationship, where a class contains an object of another class as a member variable. Inheritance is an "is-a" relationship, where a subclass extends a superclass to inherit its attributes and methods.

Ans.4

An abstract class is a class that cannot be instantiated and is used as a base class for other classes to inherit from. It can contain abstract methods, which are declared but not implemented in the abstract class and must be implemented in the subclasses.

Ans.5

Constructor is used for initialising the object state whereas method is used for exposing the object's behaviour. Constructors have no return type but Methods should have a return type. Even if it does not return anything, the return type is void. If the constructor is not defined, then a default constructor is provided by the java compiler. The constructor name should be equal to the class name. A constructor cannot be marked as final because whenever a class is inherited, the constructors are not inherited. A method can be defined as final but it cannot be overridden in its subclasses.

Ans.6

The diamond problem is an issue that can arise in programming languages that support multiple inheritance, where a class inherits from two or more classes that have a common ancestor. This can cause ambiguity in the method resolution order, leading to unpredictable behaviour. In Java, multiple inheritance is not supported directly, but it can be simulated using interfaces. A class can implement one or more interfaces, effectively inheriting their properties and methods.

Ans.7

Instance variables are accessible by all the methods in the class. They are declared outside the methods and inside the class. These variables describe the properties of an object and remain bound to it. Local variables are those variables present within a block, function, or constructor and can be accessed only inside them. The utilisation of the variable is restricted to the block scope.

Ans.8

Marker interfaces or tagging interfaces are those which have no methods and constants defined in them. They help the compiler and JVM get run time-related object information.

Data Structures and Algorithms

Ans.1

This storage area in the Java heap is specifically used to store String literals, with the aim of reducing the creation of temporary String objects through sharing. For sharing to be possible, an immutable class is required. Also no external synchronisation of threads is required if the String objects are immutable. In Hash Tables and HashMaps, keys are String objects and should thus be immutable to avoid modification.

Ans.2

If we create a String using new(), then a new object is created in the heap memory even if that value is already present in the heap memory. If we create a String using String literal and its value already exists in the string pool, then that String variable also points to that same value in the String pool without the creation of a new String with that value.

Ans.3

The Collections framework is a set of interfaces and classes that provide common data structures such as lists, sets, and maps.

Ans.4

ArrayList is a dynamic array that can grow or shrink as needed, while LinkedList is a doubly linked list that allows fast insertion and deletion of elements.

Ans.5

HashMap is a hash table that stores key-value pairs, while TreeMap is a red-black tree that stores key-value pairs in sorted order.

Ans.6

HashSet is a set that stores unique elements in an unordered manner, while TreeSet is a set that stores unique elements in a sorted manner.

Ans.7

Iterator is used to traverse a collection in a forward direction, while ListIterator is used to traverse a list in both forward and backward directions.

Ans.8

An ArrayList is a dynamic array that can grow or shrink as needed, while a LinkedList is a doubly linked list that allows fast insertion and deletion of elements. Accessing an element in an ArrayList is O(1) on average, while accessing an element in a LinkedList is O(n) on average.

Ans.9

The Comparable interface is used to provide a natural ordering for a class. It contains a single method compareTo() that compares the current object with another object of the same class and returns a negative integer, zero, or a positive integer depending on whether the current object is less than, equal to, or greater than the other object, respectively.

Ans.10

A HashSet is an unordered collection of unique elements, while a TreeSet is a sorted collection of unique elements. HashSet uses a hash table to store its elements, while TreeSet uses a balanced binary tree.

Ans.11

The java.util.concurrent package provides classes for concurrent programming, including thread pools, locks, atomic variables, and concurrent collections. It is designed to improve performance and scalability in multi threaded applications.

Exception Handling

Ans.1

An exception is an event that occurs during the execution of a program that disrupts the normal flow of instructions.

Ans.2

When an exception occurs, it tries to locate the matching catch block. If the matching catch block is located, then that block is executed. Otherwise the exception propagates through the method call stack and goes into the caller method where the process of matching the catch block is performed. This happens until the matching catch block is found. IIn case the match is not found, the program gets terminated in the main method.

Ans.3

Checked exceptions are checked at compile time, while unchecked exceptions are checked at runtime.

Ans.4

The try-catch block is used to handle exceptions in Java.

Ans.5

Throw is used to explicitly throw an exception, while throws is used to declare a method that can potentially throw an exception.

Ans.6

The finally block is used to execute a block of code regardless of whether an exception is thrown or not.

Ans.7

In Java, Java.lang.Throwable is the super class of all exception classes and all exception classes are derived from this base class.

Ans.8

Java Enterprise Edition (Java EE) is a set of specifications and APIs for developing enterprise applications in Java. It includes a range of technologies, such as Servlets, JSPs, EJBs, JPA, JMS, and JNDI.

Ans.9

A Servlet is a Java class that processes HTTP requests and generates HTTP responses. A JSP (JavaServer Pages) is a text-based document that is compiled into a Servlet. JSPs allow for a separation of presentation logic from business logic.

Ans.10

The Java Persistence API (JPA) is a specification for object-relational mapping (ORM) in Java. It provides a set of interfaces and annotations for mapping Java objects to relational database tables and vice versa.

Ans.11

Stateful session beans maintain a conversational state with the client, while stateless session beans do not. Stateful session beans are used for long-running conversations with a client, while stateless session beans are used for short-lived tasks.

Multithreading

Ans.1

A thread is a lightweight process that can run concurrently with other threads in a program. Java thread life cycle has 5 stages: New, Runnable, Running, Non-Runnable(Blocked/ Waiting), Terminated.

Ans.2

A process is a program in execution, while a thread is a subset of a process. Threads share memory while processes do not.

Ans.3

There are a total of 3 different types of priority available in Java. MIN_PRIORITY:Integer value 1. MAX_PRIORITY: Integer value 10. NORM_PRIORITY: Integer value 5 .

Ans.4

Context switching in Java is the process of switching from one thread to another by the operating system's scheduler. During context switching, the current thread's context, including its register values and program counter, are saved, and the next thread's context is restored.

Ans.5

In Java, user threads have a specific life cycle and its life is independent of any other thread and are used for critical tasks. Daemon threads are basically referred to as a service provider that provides services and support to user threads. JVM does not wait for daemon threads to finish their tasks before termination., but waits for user threads.

Ans.6

Synchronization is the mechanism that ensures that only one thread can access a shared resource at a time.

Ans.7

A deadlock is a situation where two or more threads are blocked waiting for each other to release the resources they need to proceed.

Ans.8

Wait() and notify() methods are used for inter thread communication in Java.

Ans.9

A process is an independent program that runs in its own memory space, while a thread is a subset of a process that runs concurrently with other threads in the same process.

Ans.10

Synchronized is used to provide exclusive access to a shared resource by allowing only one thread to access it at a time, while volatile is used to ensure visibility of changes made to a shared variable by guaranteeing that all threads see the same value.

Ans.11

The sleep() method is used to pause the execution of a thread for a specified amount of time, allowing other threads to execute in the meantime.

Ans.12

Wait() is a method of the Object class that is used to pause the execution of a thread and release the lock on an object, while sleep() is a method of the Thread class that is used to pause the execution of a thread without releasing any locks.

Ans.13

Notify() is used to wake up a single thread that is waiting on an object, while notifyAll() is used to wake up all threads that are waiting on an object.