

Freshers looking for a job in the software industry should have a good understanding of Java and its concepts. In this blog, we will outline 40 Java interview questions for freshers that are commonly asked in job interviews. These questions cover topics like object-oriented programming, data types, control structures, and exception handling.

Table of contents

1. Java Interview Questions for Freshers

- What is Java?
- What are the features of Java?
- What is JVM?
- What is the difference between JDK, JRE, and JVM?
- What is the difference between a class and an object?
- What is object-oriented programming?
- What are the four principles of OOP?

Java Interview Questions for Freshers

Before we move on to the next part, if you would like to explore JAVA through a self-paced course, you can take up [GUVI's JAVA self-paced course](#) with certification.

1. What is Java?

Java is a high-level, object-oriented programming language that was developed by Sun Microsystems in 1995. It is platform-independent,



meaning that programs written in Java can run on any platform that has a Java Virtual Machine (JVM) installed. [Visit Oracle's official website to download JVM](#)

2. What are the features of Java?

Java has several features that make it a popular programming language. Some of these features include platform independence, object-oriented programming, automatic memory management, robustness, and security.

3. What is JVM?

JVM stands for Java Virtual Machine. It is an abstract machine that provides the runtime environment in which Java programs are executed. The JVM interprets Java bytecode and translates it into machine-specific code.

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

JDK stands for Java Development Kit. It is a software development kit that includes tools for developing, compiling, and debugging Java programs. JRE stands for Java Runtime Environment. It is a software environment that provides the necessary runtime libraries and components for running Java programs. JVM is the virtual machine that executes the Java bytecode.

5. What is the difference between a class and an object?

A class is a blueprint or template for creating objects, while an object is an instance of a class. In other words, a class defines the properties and behaviours of an object, while an object is an instance of those properties and behaviours.



6. What is object-oriented programming?

Object-oriented programming (OOP) is a programming paradigm that is based on the concept of objects. In OOP, programs are organized around objects, which have properties (attributes) and behaviors (methods). OOP emphasizes encapsulation, inheritance, polymorphism, and abstraction.

7. What are the four principles of OOP?

The four principles of OOP are encapsulation, inheritance, polymorphism, and abstraction.

[What does it take to become a Java Developer in 2023?](#)

8. What is inheritance?

Inheritance is a mechanism in which one class inherits properties and methods from another class. The class that inherits is called the subclass or derived class, and the class that is inherited from is called the superclass or base class.

9. What is encapsulation?

Encapsulation is a mechanism in which data and methods are combined into a single unit called a class. Encapsulation provides data hiding, which means that the internal implementation details of a class are hidden from the outside world.

10. What is polymorphism?

Polymorphism is a mechanism in which a single method can have different forms or implementations. Polymorphism can be achieved through method overloading or method overriding.

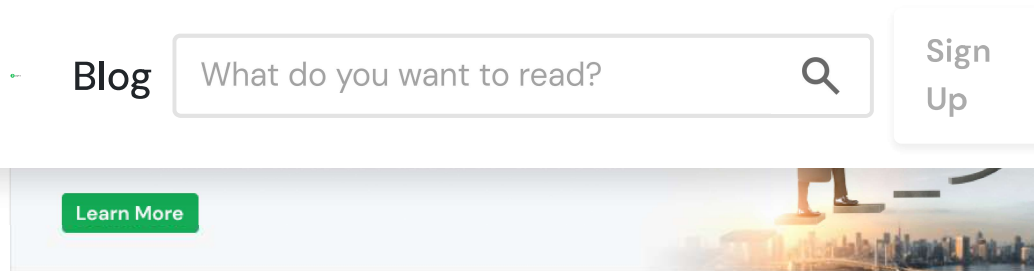
11. What is abstraction?



Abstraction is a mechanism in which complex systems can be simplified by hiding unnecessary details. In Java, abstraction can be achieved through abstract classes and interfaces.

12. What are access modifiers?

Access modifiers are keywords that are used to specify the accessibility of a class, method, or variable. There are four types of access modifiers in Java: public, private, protected, and default.



13. What are static variables and methods?

Static variables and methods belong to the class rather than to an instance of the class. This means that they can be accessed without creating an object of the class.

14. What are the final variables and methods?

Final variables and methods cannot be modified or overridden once they are defined. Final variables are constants, while final methods cannot be overridden by subclasses.

15. What is a constructor?

A constructor is a special method that is used to initialize objects. It has the same name as the class and does not have a return type.

16. What are the different types of variables in Java?



There are three types of variables in Java: local variables, instance variables, and static variables.

17. What is an array?

An array is a collection of elements of the same data type. It can be used to store data in a more organized and efficient way.

☰ Table of contents

Control structures are statements that are used to control the flow of a program. They include if-else statements, switch statements, loops (for, while, do-while), and break and continue statements.

19. What is exception handling?

Exception handling is a mechanism in which errors or exceptions that occur during the program execution are caught and handled gracefully. It involves using try-catch blocks to catch exceptions and handle them appropriately.

Further Java Interview questions will test your Logical as well

20. What are the different types of exceptions in Java?

There are two types of exceptions in Java: checked exceptions and unchecked exceptions. Checked exceptions are checked at compile time, while unchecked exceptions are checked at runtime. Examples of checked exceptions include `IOException` and `ClassNotFoundException`, while examples of unchecked exceptions include `NullPointerException` and `ArrayIndexOutOfBoundsException`.

21. What is the difference between a HashSet and a TreeSet?

A `HashSet` is an unordered collection of unique elements, while a `TreeSet` is a sorted collection of unique elements.



22. How can you implement a Stack using an Array?

You can implement a Stack using an Array by maintaining a pointer to the top of the stack and pushing and popping elements from the top of the array.

23. What is the difference between a shallow copy and a deep copy in Java?

A shallow copy creates a new object with the same reference as the original object, while a deep copy creates a new object with a new reference.

24. How do you reverse a linked list in Java?

You can reverse a linked list in Java by iterating through the list and reversing the order of the links between nodes.

25. What is the difference between a String, StringBuilder, and StringBuffer in Java?

A String is immutable, while a StringBuilder and StringBuffer are mutable. StringBuilder is not thread-safe, while StringBuffer is thread-safe.

26. What is a binary search tree? How can you implement one in Java?

A binary search tree is a data structure in which each node has at most two children, and the left child is smaller than the parent while the right child is larger. You can implement one in Java using recursion and comparison operators.

27. What is the difference between an abstract class and an interface in Java?



An abstract class is a class that cannot be instantiated and can contain abstract methods, while an interface is a collection of abstract methods and constants that can be implemented by any class.

28. How do you handle multiple exceptions in Java?

You can handle multiple exceptions in Java using a try-catch block with multiple catch clauses.

29. How do you implement a merge sort algorithm in Java?

You can implement a merge sort algorithm in Java using recursion and a merge method to combine two sorted arrays.

30. Provide an example for Polymorphism.

Polymorphism in Java refers to the ability of objects of different classes to be treated as if they are of the same class. An example is a superclass `Animal` with a method `makeSound()` that is overridden by subclasses `Dog` and `Cat`.

31. What is a lambda expression in Java? Provide an example.

A lambda expression in Java is a function that can be created without belonging to any class. An example is $(x, y) \rightarrow x + y$, which takes two integer parameters and returns their sum.

32. What is the difference between a private and a protected method in Java?

A private method is only accessible within the same class, while a protected method is accessible within the same class and any subclass.



33. How do you implement a bubble sort algorithm in Java?

You can implement a bubble sort algorithm in Java by iterating through the array and swapping adjacent elements if they are in the wrong order.

34. What is the difference between a while loop and a do-while loop in Java?

A while loop executes the loop body if the condition is true, while a do-while loop executes the loop body at least once before checking the condition.

35. How do you implement a binary search algorithm in Java?

You can implement a binary search algorithm in Java by recursively dividing the search interval in half until the target element is found.

36. What is a thread in Java? Provide an example of how to create a thread.

A thread in Java is a lightweight process that can execute independently of other threads. An example of creating a thread is extending the Thread class or implementing the Runnable interface and calling the start() method.

37. What is the difference between a static and non-static variable in Java?

A static variable is associated with the class, while a non-static variable is associated with an instance of the class.

38. What is a deadlock in Java? How can you prevent it?



A deadlock in Java occurs when two or more threads are blocked waiting for each other to release resources. You can prevent it by avoiding circular dependencies and using thread-safe synchronization.

39. What is an interface in Java? Provide an example.

An interface in Java is a collection of abstract methods and constants that can be implemented by any class. An example is the Comparable interface, which defines a method compareTo() for comparing objects.



40. How do you implement a quick sort algorithm in Java?

You can implement a quick sort algorithm in Java using recursion and a partition method to divide the array into two parts.

if you would like to explore JAVA through a self-paced course, you can take up [GUVI's JAVA self-paced course](#) with certification.

Career transition

