

Java Interview Questions and Answers 2023

Most Asked Java Interview Questions & Answers

Java Interview Questions and Answers are given on this page for technical interview preparation. You can practice most asked java questions 2022-23 for freshers. We have covered java topics like **Encapsulation, Inheritance, Special Operator** etc.

Page Highlights:

- What is Java?
- Top 45 Java Interview Questions
- Technical Interview Questions



▶ Table of Contents

Introduction to Java

Java is a widely used programming language expressly designed for use in the distributed environment of the internet. Java can be used to create complete applications that may run on a single computer or be distributed among servers and clients in a network.

Features of JAVA:-

- Object Oriented
- Platform Independent
- Portable
- Robust
- High Performance
- Dynamic

Top 45 Java Interview Questions

What is a class?

Ans. Class is a blueprint or template from which objects are created and describes the contents of the object. A class should be defined before creating an object.

What are the different types of memory areas allocated by JVM?

Ans.

- **Class Memory:** Stores method data, fields, method codes etc.,
- **Heap Memory:** Memory is allocated to store objects.
- **Stack Memory:** Memory is allocated to store local variables and order of method execution.
- **Program Counter:** This is a register(PC), that holds the address of instruction that is currently being executed.
- **Native Method Stack:** Contains all native methods.

State the difference between heap memory and stack memory.

Ans. Heap Memory:

- It is used to store objects.
- If heap memory gets filled then it returns `java.lang.OutOfMemoryError`.



- Accessing this memory is slower when compared to stack.
- Heap space is used throughout the applications.

Stack Memory :

- It is used to store the order of method execution and local variables.
- If stack memory gets filled then it returns `java.lang.StackOverflowError`.
- Accessing this memory is faster when compared to heap.
- Stack space is used for methods that are currently running.

What is object cloning?

What are the access specifiers in Java?

Is Java 100% object oriented?

What is JRE?

Ans. JRE means Java Runtime Environment. It is a software package that contains JVM(Java Virtual Machine), class libraries and other components.

What is JDK?

What is a variable in Java?

Ans. Variable is nothing but a location of memory to store data and the data can be accessed by using the variable that is assigned to. Variables in Java are case-sensitive.

Example : `int i = 90;` Int is the datatype, i is the variable and 90 is the data.

What is meant by statically-typed language?

Ans. Java is a statically-typed language. Statically-typed language means all the variables must be declared before they are used.

Can we write static public void instead of public static void?

Ans. Yes, we can write static public void instead of public static void. The program compiles with no errors. Because the order of specifier's doesn't matter.

Explain the “InstanceOf” operator in Java.

Ans. “InstanceOf” operator is a type comparison operator which compares an object to a specified type. It can be explained by an example.

Example:

```
Class A{
    public static void main(String args[]){
        String t = "asd";
        Boolean r;
        r = t instanceof String;
        System.out.println("Is t is object of string?" +r);
    }
}
```

Output : true

Explain about the CopyOfRange() method in Java.

Ans. CopyOfRange() method is used to copy elements from one array to another array. It can be explained by an example.

Syntax : CopyOfRange(Source Array, start_index, end_index)

Example:

```
Class A{
    public static void main(String args[]){
        int[] s = {2,3,12,4,12,-2};

        int[] d1 = Arrays.copyOfRange(s,2,5);

        System.out. println("d1= "+Arrays.toString(d1));
    }
}

Output : {12,4,12}
```

What is the gc() method?

Ans. To invoke Garbage Collector, this gc() method is called. It is found in both system and runtime classes. It automatically releases the memory when an object is no longer used.

Example : public static void gc()

What are constructors in Java?

Ans:- The purpose of the default constructor is to assign the default value to the objects. The java compiler creates a default constructor implicitly if there is no constructor in the class.

Example:

```
public class Student3{  
    int id;  
    String name;  
  
    void display(){System.out.println(id+" "+name);}  
  
    public static void main(String args[]){  
  
        Student3 s1=new Student3();  
  
        Student3 s2=new Student3();  
  
        s1.display();  
  
        s2.display();  
  
    }  
}  
  
Output:  
0 null  
0 null
```

Can we assign the reference to this variable?

Ans. No, this cannot be assigned to any value because it always points to the current class object and this is the final reference in Java. However, if we try to do so, the compiler error will be shown. Consider the following example.

```

public class Test{
    public Test(){
        this = null;
        System.out.println("Test class constructor called");
    }

    public static void main (String args[]){
        Test t = new Test();
    }
}

Output : Test.java:5: error: cannot assign a value to final variable this
        this = null;
               ^
1 error

```

What are the advantages of passing this into a method instead of the current class object itself?

Ans. As we know, that this refers to the current class object, therefore, it must be similar to the current class object. However, there can be two main advantages of passing this into a method instead of the current class object.

- This is a final variable. Therefore, this cannot be assigned to any new value whereas the current class object might not be final and can be changed.
- This can be used in the synchronized block.

What is the Inheritance?

Ans. Inheritance is a mechanism by which one object acquires all the properties and behavior of another object of another class. It is used for Code Reusability and Method Overriding. The idea behind inheritance in Java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can

reuse methods and fields of the parent class. Moreover, you can add new methods and fields in your current class also. Inheritance represents the IS-A relationship which is also known as a parent-child relationship.

There are five types of inheritance in Java.

- Single-level inheritance
- Multi-level inheritance
- Multiple Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

Why is Inheritance used in Java?

Ans. There are various advantages of using inheritance in Java that is given below.

- Inheritance provides code reusability. The derived class does not need to redefine the method of base class unless it needs to provide the specific implementation of the method.
- Runtime polymorphism cannot be achieved without using inheritance.
- We can simulate the inheritance of classes with the real-time objects which makes OOPs more realistic.
- Inheritance provides data hiding. The base class can hide some data from the derived class by making it private.
- Method overriding cannot be achieved without inheritance. By method overriding, we can give a specific implementation of some basic method contained by the base class.

What is aggregation?

Ans. Aggregation can be defined as the relationship between two classes where the aggregate class contains a reference to the class it owns. Aggregation is best described as a has-a relationship. For example, The aggregate class Employee having various fields such as age, name, and salary also contains an object of Address class having various fields such as Address-Line 1, City, State, and pin-code. In other words, we can say that Employee (class) has an object of Address class.

What is super in java?

Ans. The super keyword in Java is a reference variable that is used to refer to the immediate parent class object. Whenever you create the instance of the subclass, an instance of the parent class is created implicitly which is referred by super reference variable. The super() is called in the class constructor implicitly by the compiler if there is no super or this.

```

public class Animal{
Animal(){

    System.out.println("animal is created");

}

class Dog extends Animal{

Dog(){

    System.out.println("dog is created");

}

}

class TestSuper4{

public static void main(String args[]){

Dog d=new Dog();

}

}

Output:

animal is created

dog is created

```

What is method overloading?

Ans. Method overloading is the polymorphism technique which allows us to create multiple methods with the same name but different signatures. We can achieve method overloading in two ways.

- Changing the number of arguments.
- Changing the return type.

What is method overriding?

Ans. If a subclass provides a specific implementation of a method that is already provided by its parent class, it is known as Method Overriding. It is used for runtime polymorphism and to implement the interface methods. Rules for Method overriding

- The method must have the same name as in the parent class.
- The method must have the same signature as in the parent class.
- Two classes must have an IS-A relationship between them.

Can we change the scope of the overridden method in the subclass?

Ans. Yes, we can change the scope of the overridden method in the subclass. However, we must notice that we cannot decrease the accessibility of the method. The following point must be taken care of while changing the accessibility of the method.

- The private can be changed to protected, public, or default.
- The protected can be changed to public or default.
- The default can be changed to public.

- The public will always remain public.

Can we modify the throws clause of the superclass method while overriding it in the subclass?

Ans. Yes, we can modify the throws clause of the superclass method while overriding it in the subclass. However, there are some rules which are to be followed while overriding in case of exception handling.

- If the superclass method does not declare an exception, a subclass overridden method cannot declare the checked exception, but it can declare the unchecked exception.
- If the superclass method declares an exception, a subclass overridden method can declare the same, subclass exception or no exception but cannot declare parent exception.

What is the final variable?

Ans. In Java, the final variable is used to restrict the user from updating it. If we initialize the final variable, we can't change its value. In other words, we can say that the final variable once assigned to a value, can never be changed after that. The final variable which is not assigned to any value can only be assigned through the class constructor.

What is the final class?

Ans. If we make any class final, we can't inherit it into any of the subclasses.

```
final class Bike{}
class Honda1 extends Bike{
void run(){System.out.println("running safely with 100kmph");}

public static void main(String args[]){
    Honda1 honda= new Honda1();
    honda.run();
}
}
```

What is the difference between the final method and abstract method?

Ans. The main difference between the final method and abstract method is that the abstract method cannot be final as we need to override them in the subclass to give its definition.

Why is the main method static?

Ans. Because the object is not required to call the static method. If we make the main method non-static, JVM will have to create its object first and then call main() method which will lead to the extra memory allocation.

Can you declare the main method as final?

Ans. Yes, We can declare the main method as public static final void main(String[] args){}.

Can we have multiple public classes in a java source file?

Ans. We can't have more than one public class in a single java source file. A single source file can have multiple classes that are not public.

What are access modifiers?

Ans. Java provides access control through public, private, and protected access modifier keywords. When none of these are used, it's called the default access modifier. A java class can only have a public or default access modifier. Read Java Access Modifiers to learn more about these in detail.

What is a multi-catch block in java?

Ans. Java 7 includes many improvements and one of the improvements was a multi-catch block where we can catch multiple exceptions in a single catch block. This makes our code shorter and cleaner when every catch block has a similar code. If a catch block handles multiple exceptions, you can separate them using a pipe (|) and in this case, the exception parameter (ex) is final, so you can't change it.

What do you know about Interface in Java?

Ans. A Java interface is a template that has only method declarations and not method implementations. It is a workaround for achieving multiple inheritances in Java. Some worth remembering important points regarding Java interfaces are:

- A class that implements the interface must provide an implementation for all methods declared in the interface.
- All methods in an interface are internally public abstract void.
- All variables in an interface are internally public static final.
- Classes do not extend but implement interfaces.

How is an Abstract class different from an Interface?

Ans.

There are several differences between an Abstract class and an Interface in Java, summed up as follows:

- Constituents – An abstract class contains instance variables, whereas an interface can contain only constants.
- Constructor and Instantiation – While an interface has neither a constructor nor it can be instantiated, an abstract class can have a default constructor that is called whenever the concrete subclass is instantiated.
- Implementation of Methods – All classes that implement the interface need to provide an implementation for all the methods contained by it. A class that extends the abstract class, however, doesn't require implementing all the methods contained in it. Only abstract methods need to be implemented in the concrete subclass.
- Type of Methods – Any abstract class has both abstract as well as non-abstract methods. Interface, on the other hand, has only a single abstract method.

Why do we use the yield() method?

Ans. The yield() method belongs to the thread class. It transfers the currently running thread to a runnable state and also allows the other threads to execute. In other words, it gives equal priority threads a chance to run. Because yield() is a static method, it does not release any lock.

Can you explain the thread lifecycle in Java?

Ans.

The thread lifecycle has the following states and follows the following order:

- **New** – In the very first state of the thread lifecycle, the thread instance is created, and the start() method is yet to be invoked. The thread is considered alive now.
- **Runnable** – After invoking the start() method, but before invoking the run() method, a thread is in the runnable state. A thread can also return to the runnable state from the waiting or sleeping state.
- **Running** – The thread enters the running state after the run() method is invoked. This is when the thread begins execution.
- **Non-Runnable** – Although the thread is alive, it is not able to run. Typically, it returns to the runnable state after some time.
- **Terminated** – The thread enters the terminated state once the run() method completes its execution. It is not alive now.

How do you make a thread stop in Java?

Ans.

There are three methods in Java to stop the execution of a thread:

- **Blocking** – This method is used to put the thread in a blocked state. The execution resumes as soon as the condition of the blocking is met. For instance, the ServerSocket.accept() is a blocking method that listens for incoming socket connections and resumes the blocked thread only when a connection is made.
- **Sleeping** – This method is used for delaying the execution of the thread for some time. A thread upon which the sleep() method is used is said to enter the sleep state. It enters the runnable state as soon as it wakes up i.e., the sleep state is finished. The time for which the thread needs to enter the sleep state is mentioned inside the braces of the sleep() method. It is a static method.
- **Waiting** – Although it can be called on any Java object, the wait() method can only be called from a synchronized block.

How does the throw keyword differ from the throws keyword?

Ans. While the throws keyword allows declaring an exception, the throw keyword is used to explicitly throw an exception. Checked exceptions can't be propagated with throw only, but throws allow doing so without the need for anything else. The throws keyword is followed by a class, whereas the throw keyword is followed by an instance. The throw keyword is used within the method, but the throws keyword is used with the method signature. Furthermore, it is not possible to throw multiple exceptions, but it is possible to declare multiple exceptions.

What do you mean by Collections in Java? What are the constituents of Collections in Java?

Ans. A group of objects in Java is known as collections. Java.util package contains, along with the date and time facilities, internationalization, legacy collection classes, etc., the various classes and interfaces for collecting. Alternatively, collections can be considered as a framework designed for storing the objects and manipulating the design in which the objects are stored. You can use collections to perform the following operations on objects:

- Deletion
- Insertion
- Manipulation
- Searching
- Sorting

Following are the various constituents of the collections framework:

- Classes – Array List, Linked List, Lists, and Vector.
- Interfaces – Collection, List, Map, Queue, Set, Sorted Map, and Sorted Set.

- Maps – HashMap, HashTable, LinkedHashMap, and TreeMap.
- Queues – Priority Queue.
- Sets – Hash Set, Linked Hash Set, and Tree Set.

What do you mean by Priority Queue in Java?

Ans. Priority queue, like a regular queue, is an abstract data type except having a priority associated with each element contained by it. The element with the high priority is served before the element with low priority in a priority queue. Elements in a priority queue are ordered either according to the comparator or naturally. The order of the elements in a priority queue represents their relative priority.

What is a package in Java? List down various advantages of packages.

Ans. Packages in Java, are the collection of related classes and interfaces which are bundled together. By using packages, developers can easily modularize the code and optimize its reuse. Also, the code within the packages can be imported by other classes and reused. Below I have listed down a few of its advantages:

- Packages help in avoiding name clashes.
- They provide easier access control on the code.
- Packages can also contain hidden classes which are not visible to the outer classes and only used within the package.
- Creates a proper hierarchical structure which makes it easier to locate the related classes.

Can we modify the throws clause of the superclass method while overriding it in the subclass?

Ans. Yes, we can modify the throws clause of the superclass method while overriding it in the subclass. However, there are some rules which are to be followed while overriding in case of exception handling.

- If the superclass method does not declare an exception, the subclass overridden method cannot declare the checked exception, but it can declare the unchecked exception.
- If the superclass method declares an exception, the subclass overridden method can declare the same, subclass exception, or no exception but cannot declare parent exception.

Can we change the scope of the overridden method in the subclass?

Ans. Yes, we can change the scope of the overridden method in the subclass. However, we must notice that we cannot decrease the accessibility of the method. The following point must be taken care of while changing the accessibility of the method.

- The private can be changed to protected, public, or default.
- The protected can be changed to public or default.
- The default can be changed to public.
- The public will always remain public.

What are the advantages of passing this into a method instead of the current class object itself?

Ans. As we know, this refers to the current class object, therefore, it must be similar to the current class object. However, there can be two main advantages of passing this into a method instead of the current class object.

- ‘this’ is a final variable. Therefore, this cannot be assigned to any new value whereas the current class object might not be final and can be changed.
- ‘this’ can be used in the synchronized block.

FAQs on Java Interview Questions

What is an object?

Ans:- The Object is the real-time entity having some state and behavior. In Java, Object is an instance of the class having the instance variables as the state of the object and the methods as the behavior of the object. The object of a class can be created by using the new keyword.

Why Java is platform independent?

Ans:- Java is called platform independent because of its byte codes which can run on any system irrespective of its underlying operating system.

[Login/Signup to comment](#)

| Support | Companies | | All Exams Dashboards | Get In Touch | Get In Touch |
|----------------------|------------------|-----------|-----------------------------|---------------------|----------------------|
| Contact Us | Accenture | Microsoft | CoCubes Dashboard | Instagram | |
| About Us | Cognizant | TCS | eLitmus Dashboard | Linkedin | support@preinsta.com |
| Refund Policy | MindTree | Infosys | HirePro Dashboard | Youtube | a.com |
| Privacy Policy | VMware | Oracle | MeritTrac Dashboard | Telegram | +91-8448440710 |
| Services | CapGemini | HCL | Mettl Dashboard | facebook | Text us on |
| Disclaimer | Deloitte | TCS Ninja | DevSquare Dashboard | Twitter | Whatsapp/Instagram |
| Terms and Conditions | Wipro | IBM | | | |

PreInsta.com

No.1 and most visited website for Placements in India.

We help students to prepare for placements with the best study material, online classes, Sectional Statistics for better focus and Success stories & tips by Toppers on PreInsta.

© 2022 Prep Insta

[Privacy Policy](#) | Copyright © 2022 Prep Insta