

# Frequently Asking Core Java Interview Questions Updated

Last Updated On: October 28, 2019 By Softwaretestingo Admin

---

## 1. Core Java Interview Questions

### Core Java Interview Questions

#### How many methods are there in object class?

Ans: 11 methods, below is the list

- protected Object clone() throws CloneNotSupportedException – Creates and returns a copy of this object.
- Public boolean equals(Object obj) – Indicates whether some other object is “equal to” this one.
- Protected void finalize() throws Throwable – Called by the garbage collector on an object when garbage collection determines that there are no more references to the object.
- public final Class getClass() – Returns the runtime class of an object.
- public int hashCode() – Returns a hash code value for the object.
- Public final void notify() – Wakes up a single thread that is waiting on this object's monitor.
- Public final void notifyAll() – Wakes up all threads that are waiting on this object's monitor.
- Public String toString() – Returns a string representation of the object.
- Public final void wait() – Causes current thread to wait until another thread invokes the notify() method or the notifyAll() method for this object.
- public final void wait(long timeout)
- public final void wait(long timeout, int Nanos)

#### Hashcode method return type?

Ans: integer

#### what is the finalize method and when it use?

Ans: Garbage collection, it executed before garbage collector runs.

#### What is equals methods?

Ans: it is used in 2 places, one is a string class, and another is an object class. In string class, it is overridden. In object class, it checks the addresses, and in string class, it checks the contents.

#### What is the difference between equal and hashCode?

Ans: hashCode can be the same; it is used internally to find out the uniqueness of an object.

Kiran — 89

Naikr — 89

suppose k=23,i=4

Then both Kiran and Naikr have the same hashCode, but both are

not the same. But if equals return same, then there must be the same hashcode.

## Core Java OOPS Interview Questions

### is it compulsory to write body in a method?

Ans: no, ex-abstract class, interface

### why do we need to learn encapsulation?

Ans: binding data variables and method in a single unit is called encapsulation. We are using encapsulation to protect our class from others so that others can not extract our class. Her variables should be private and method some time public so that our requirement solved properly.

### Difference between encapsulation and abstraction?

Ans: in abstraction, we expose only the required things. But in encapsulation, we are binding the data properly.

### What is polymorphism?

Ans: behaving differently at different time by the same entity. It can achieve in 2 ways. Overloading and overriding. in overloading this happens and in overriding this happen.

### Why are you using overriding?

Ans: To change the implementation of the parent class.

### Why we use interface?

Ans: if we want 100% abstraction then we should go with an interface. To maintain standards.

### Do you need to implement all methods of interface?

Ans: if you don't want to implement all methods, then you have to make that class as abstract.

### What is the use of an abstract class?

Ans: if you want to implement or expose some methods and don't want to implement some methods then go for abstract class

### List l=new ArrayList(); and ArrayList al=new ArrayList(); which is good and why ?

Ans: List l=new ArrayList() is good.

## what is the difference between final and static?

Ans: in final, the value can not be changed, but in static, you can change the value.

## What is difference between JDK,JRE and JVM?

Ans:

**JVM:** JVM is an acronym for Java Virtual Machine, it is an abstract machine which provides the runtime environment in which Java bytecode can be executed. It is a specification. JVMs are available for many hardware and software platforms (so JVM is platform dependent).

**JRE:** JRE stands for Java Runtime Environment. It is the implementation of JVM.

**JDK:** JDK is an acronym for Java Development Kit. It physically exists. It contains JRE + development tools.

## How many types of memory areas are allocated by JVM?

Ans: Many types:

- Class(Method) Area
- Heap
- Stack
- Program Counter Register
- Native Method Stack

## What is JIT compiler?

Ans: Just-In-Time(JIT) compiler: It is used to improve performance. JIT compiles parts of the bytecode that have similar functionality at the same time, and hence reduces the amount of time needed for compilation. Here the term “compiler” refers to a translator from the instruction set of a Java virtual machine (JVM) to the instruction set of a specific CPU.

## What is a platform?

Ans: A platform is the hardware or software environment in which a program runs. There are two types of platforms software-based and hardware-based. Java provides the software-based platform.

## What is the main difference between the Java platform and other platforms?

Ans: The Java platform differs from most other platforms in the sense that it's a software-based platform that runs on top of other hardware-based platforms. It has two components:

- Runtime Environment
- API(Application Programming Interface)

## What gives Java its 'write once and run anywhere' nature?

Ans: The bytecode. Java is compiled to be a bytecode which is the intermediate language between source code and machine code. This bytecode is not platform specific and hence can be fed to any platform.

## What is classloader?

Ans: The classloader is a subsystem of JVM that is used to load classes and interfaces. There are many types of classloaders, e.g. Bootstrap classloader, Extension classloader, System classloader, Plugin classloader etc.

## Read Also: Java Collection Framework Interview Questions

## Is Empty .java file name a valid source file name?

Ans: Yes, save your java file by .java only, compile it by javac .java and run by java your class name. Let's take a simple example:

```
//save by .java only
class A{
public static void main(String args[]){
System.out.println("Hello java");
}
}
//compile by javac .java
//run by java A
compile it by javac .java
```

Ans: run it by java A

## If I don't provide any arguments on the command line, then the String array of Main method will be empty or null?

Ans: It is empty. But not null.

## What if I write static public void instead of the public static void?

Ans: Program compiles and runs properly.

## What is the default value of the local variables?

Ans: The local variables are not initialized to any default value, neither primitives nor object references.

## What is the difference between an object-oriented programming language and object-based programming

## language?

Ans: Object-based programming languages follow all the features of OOPs except Inheritance. Examples of object-based programming languages are JavaScript, VBScript etc.

## What will be the initial value of an object reference which is defined as an instance variable?

Ans: The object references are all initialized to null in Java.

## What is a constructor?

Ans: Constructor is just like a method that is used to initialize the state of an object. It is invoked at the time of object creation.

## What is the purpose of a default constructor?

Ans: The default constructor provides the default values for the objects. The java compiler creates a default constructor only if there is no constructor in the class.

## Does constructor return any value?

Ans: yes, that is a current instance (You cannot use return type yet it returns a value).

## Is constructor inherited?

Ans: No, the constructor is not inherited.

## Can you make a constructor final?

Ans: No, constructor can't be final.

## What is a static variable?

Ans: static variable is used to refer the common property of all objects (that is not unique for each object), e.g. company name of employees, college name of students etc. static variable gets memory only once in a class area at the time of class loading.

## What is the static method?

Ans: A static method belongs to the class rather than an object of a class. A static method can be invoked without the need for creating an instance of a class. the static method can access static data member and can change the value of it.

## Why is the main method static?

Ans: because the object is not required to call the static method if It were a non-static method, JVM creates the object first then call

the main() method that will lead to the problem of extra memory allocation.

### What is a static block?

Ans: Is used to initialize the static data member. It is executed before the main method at the time of classloading.

### Can we execute a program without main() method?

Ans: Yes, one of the ways is a static block.

### What if the static modifier is removed from the signature of the main method?

Ans: Program compiles. But at runtime throws an error "NoSuchMethodError".

### What is the difference between static (class) method and instance method?

Ans: static or class method instance method

Static Method	Instance Method
A method, i.e. declared as static is known as a static method.	A method, i.e. not declared as static, is known as an instance method.
The object is not required to call the static method.	The object is required to call instance methods.
Non-static (instance) members cannot be accessed in a static context (static method, static block, and static nested class) directly.	Static and non-static variables can be accessed in instance methods.

For example: public static int  
cube(int n){ return n\*n\*n;}

For example: public void msg()  
{...}.

### **What is this in java?**

Ans: It is a keyword that refers to the current object.

### **What is Inheritance?**

Ans: Inheritance is a mechanism by which one object acquires all the properties and behaviour of another object in another class. It represents the IS-A relationship. It is used for Code Reusability and Method Overriding.

### **Which class is the superclass of every class?**

Ans: Object class.

### **Why are multiple inheritances not supported in java?**

Ans: To reduce the complexity and simplify the language, multiple inheritances are not supported in java in case of class.

### **What is composition?**

Ans: Holding the reference of the other class within some other class is known as composition.

### **What is the difference between aggregation and composition?**

Ans: Aggregation represents a weak relationship, whereas composition represents a strong relationship. For example, the bike has an indicator (aggregation), but the bike has an engine (composition).

### **Why does Java not support pointers?**

Ans: Pointer is a variable that refers to the memory address. They are not used in Java because they are unsafe(unsecured) and complex to understand.

### **What is super in java?**

Ans: It is a keyword that refers to the immediate parent class object.

### **Can you use this() and super() both in a constructor?**

Ans: No. Because super() or this() must be the first statement.



### What is object cloning?

Ans: The object cloning is used to create the exact copy of an object.

### What is method overloading?

Ans: If a class have multiple methods by the same name but different parameters, it is known as Method Overloading. It increases the readability of the program.

### Why is method overloading not possible by changing the return type in java?

Ans: Because of ambiguity.

### Can we overload the main() method?

Ans: Yes, You can have many main() methods in a class by overloading the main method.

### 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 provide the specific implementation of the method.

### Can we override the static method?

Ans: No, you can't override the static method because they are the part of class not object.

### Why can we not override static method?

Ans: It is because the static method is the part of a class and it is bound with class whereas instance method is bound with an object and static gets memory in the class area, and instance gets memory in a heap.

### Can we override the overloaded method?

Ans: Yes.

### Difference between method Overloading and Overriding.

Ans:

Method Overloading	Method Overriding
Method overloading increases	Method overriding provides the



the readability of the program.	specific implementation of the method that is already provided by its superclass.
Method overloading occurs within the class.	Method overriding occurs in two classes that have an IS-A relationship.
In this case, a parameter must be different.	In this case, a parameter must be the same.

### **What is the final variable?**

Ans: If you make any variable as final, you cannot change the value of a final variable(It will be constant).

### **What is the final method?**

Ans: Final methods can't be overridden.

### **What is the final class?**

Ans: Final class can't be inherited.

### **What is the final blank variable?**

Ans: A final variable, not initialized at the time of declaration, is known as a final blank variable.

### **Can we initialize the final blank variable?**

Ans: Yes, only in the constructor if it is non-static. If it is static blank final variable, it can be initialized only in the static block.

### **Can you declare the main method as final?**

Ans: Yes, such as `public static final void main(String[] args){}`.

### **What is Runtime Polymorphism?**

Ans: Runtime polymorphism or dynamic method dispatch is a process in which a call to an overridden method is resolved at

runtime rather than at compile-time.

In this process, an overridden method is called through the reference variable of a superclass. The determination of the method to be called is based on the object being referred to by the reference variable.

### **Can you achieve Runtime Polymorphism by data members?**

Ans: No.

### **What is the difference between static binding and dynamic binding?**

Ans: In case of a static binding type of object is determined at compile time whereas in a dynamic binding type of object is determined at runtime.

### **What is abstraction?**

Ans: Abstraction is a process of hiding the implementation details and showing only functionality to the user.

Abstraction lets you focus on what the object does instead of how it does it.

### **What is the difference between abstraction and encapsulation?**

Ans: Abstraction hides the implementation details, whereas encapsulation wraps code and data into a single unit.

### **What is an abstract class?**

Ans: A class that is declared as abstract is known as an abstract class. It needs to be extended, and its method implemented. It cannot be instantiated.

### **Can there be any abstract method without an abstract class?**

Ans: No, if there is an abstract method in a class, that class must be abstract.

### **Can you use abstract and final both with a method?**

Ans: No, because the abstract method needs to be overridden, whereas you can't override a final method.

### **Is it possible to instantiate the abstract class?**

Ans: No, the abstract class can never be instantiated.

## **What is an interface?**

Ans: Interface is a blueprint for a class that has static constants and abstract methods. It can be used to achieve full abstraction and multiple inheritances.

## **Can you declare an interface method static?**

Ans: No, because methods of an interface are abstract by default, and static and abstract keywords can't be used together.

## **Can an Interface be final?**

Ans: No, because another class provides its implementation.

## **What is marker interface?**

Ans: An interface that has no data member and method is known as a marker interface. For example, Serializable, Cloneable etc.

## **What is the difference between abstract class and interface?**

Ans: Abstract class Interface

- An abstract class can have a method body (non-abstract methods). The interface has only abstract methods.
- An abstract class can have instance variables. An interface cannot have instance variables.
- An abstract class can have a constructor. An interface cannot have a constructor.
- An abstract class can have static methods. An interface cannot have static methods.
- You can extend one abstract class. You can implement multiple interfaces.

## **Can we define private and protected modifiers for variables in interfaces?**

Ans: No, they are implicitly public.

## **When can an object reference be cast to an interface reference?**

Ans: An object reference can be cast to an interface reference when the object implements the referenced interface.

## **What is a package?**

Ans: A package is a group of similar type of classes interfaces and sub-packages. It provides access protection and removes naming collision.

## **Do I need to import java.lang package any time? Why?**

Ans: No. It is by default loaded internally by the JVM.

## **Can I import the same package/class twice? Will the JVM load the package twice at runtime?**

Ans: One can import the same package or the same class multiple

times. Neither compiler nor JVM complains about it. But the JVM will internally load the class only once no matter how many times you import the same class.

### What is a static import?

Ans: By static import, we can access the static members of a class directly, there is no to qualify it with the class name.

### What is Exception Handling?

Ans: Exception Handling is a mechanism to handle runtime errors. It is mainly used to handle checked exceptions.

### What is the difference between Checked Exception and Unchecked Exception?

Ans:

Checked Exception	Unchecked Exception
The classes that extend Throwable class except RuntimeException and Error are known as checked exceptions e.g. IOException, SQLException etc. Checked exceptions are checked at compile-time.	The classes that extend RuntimeException are known as unchecked exceptions, e.g. ArithmeticException, NullPointerException etc. Unchecked exceptions are not checked at compile-time.

### What is the base class for Error and Exception?

Ans: Throwable.

### Is it necessary that a catch block must follow each try block?

Ans: It is not necessary that a catch block must follow each try block. It should be followed by either a catch block OR a finally

block. And whatever exceptions are likely to be thrown should be declared in the throws clause of the method.

### What is finally block?

Ans: finally block is a block that is always executed.

### Can finally block be used without a catch?

Ans: Yes, by try block. finally must be followed by either try or catch.

### Is there any case when finally will not be executed?

Ans: finally will not be executed if program exits(either by calling System.exit() or by causing a fatal error that causes the process to abort).

### What is the difference between throw and throws?

Ans:

Throw Keyword	Throws Keyword
throw is used to throw an exception explicitly.	Throws are used to declare an exception.
Checked exceptions cannot be propagated with throw only.	A checked exception can be propagated with throws.
An instance follows the throw.	Throws are followed by class.
the throw is used within the method.	Throws are used with the method signature.
You cannot throw multiple exceptions	You can declare multiple exceptions, e.g. public void

	method()throws IOException, SQLException.
--	--

### Can an exception be rethrown?

Ans: Yes.

### Can subclass overriding method declare an exception if the parent class method doesn't throw an exception?

Ans: Yes, but only unchecked exception not checked.

### What is exception propagation?

Ans: Forwarding the exception object to the invoking method is known as exception propagation.

### What is the meaning of immutable in terms of String?

Ans: The simple meaning of **immutable** is unmodifiable or unchangeable. Once the string object has been created, its value can't be changed.

### Why are string objects immutable in java?

Ans: Because Java uses the concept of a string literal. Suppose there are 5 reference variables, all refer by one object "Sachin". If one reference variable changes the value of the object, it will be affected by all the reference variables. That is why string objects are immutable in java.

### How many ways can we create the string object?

Ans: There are two ways to create the string object, by a string literal and by new keyword.

### How many objects will be created in the following code?

Ans:

```
String s1="Welcome";
```

```
String s2="Welcome";
```

```
String s3="Welcome";
```

Only one object.

### Why java uses the concept of a string literal?

Ans: To make Java more memory efficient (because no new

objects are created if it exists already in constant string pool).

### **How many objects will be created in the following code?**

Ans: String s = new String("Welcome");

Two objects, one in constant string pool and other in non-pool(heap).

### **What is the basic difference between string and StringBuffer object?**

Ans: String is an immutable object. StringBuffer is a mutable object.

### **What is the difference between StringBuffer and StringBuilder?**

Ans: StringBuffer is synchronized whereas StringBuilder is not synchronized.

### **How can we create an immutable class in java?**

Ans: We can create an immutable class as the String class by defining a final class and

### **What is the purpose of the toString() method in java?**

Ans: The toString() method returns the string representation of an object. If you print any object, java compiler internally invokes the toString() method on the object. So overriding the toString() method, returns the desired output, it can be the state of an object etc. depends on your implementation.

### **What is a nested class?**

Ans: A class which is declared inside another class is known as a nested class. There are 4 types of nested class member inner class, local inner class, anonymous inner class, and static nested class.

### **Is there any difference between nested classes and inner classes?**

Ans: Yes, inner classes are non-static nested classes, i.e. inner classes, are the part of nested classes.

### **Can we access the non-final local variable, inside the local inner class?**

Ans: No, the local variable must be constant if you want to access it in a local inner class.



### What is a nested interface?

Ans: Any interface, i.e. declared inside the interface or class, is known as a nested interface. It is static by default.

### Can a class have an interface?

Ans: Yes, it is known as the nested interface.

### Can an Interface have a class?

Ans: Yes, they are static implicitly.

### What is Garbage Collection?

Ans: Garbage collection is a process of reclaiming the unused runtime objects. It is performed for memory management.

### What is GC()?

Ans: GC() is a daemon thread. GC() method is defined in System class that is used to send the request to JVM to perform garbage collection.

### What is the purpose of the finalize() method?

Ans: finalize() method is invoked just before the object is garbage collected. It is used to perform cleanup processing.

### Can unreferenced objects be referenced again?

Ans: Yes.

### What kind of thread is the Garbage collector thread?

Ans: Daemon thread.

### What is the difference between final, finally and finalize?

Ans:

- **Final:** final is a keyword, final can be variable, method or class. You, can't change the value of the final variable, can't override the final method, can't inherit the final class.
- **Finally:** finally block is used in exception handling. Finally, the block is always executed.
- **finalize():** finalize() method is used in garbage collection. finalize() method is invoked just before the object is garbage collected. The finalize() method can be used to perform any cleanup processing.

### What is the purpose of the Runtime class?

Ans: The purpose of the Runtime class is to provide access to the Java runtime system.

### How will you invoke any external process in Java?

Ans: By Runtime.getRuntime().exec(?) method.

## **What is the difference between the Reader/Writer class hierarchy and the InputStream/OutputStream class hierarchy?**

Ans: The Reader/Writer class hierarchy is character-oriented, and the InputStream/OutputStream class hierarchy is byte-oriented.

## **What an I/O filter?**

Ans: An I/O filter is an object that reads from one stream and writes to another, usually altering the data in some way as it is passed from one stream to another.

## **What is serialization?**

Ans: Serialization is a process of writing the state of an object into a byte stream. It is mainly used to travel object's state on the network.

## **What is Deserialization?**

Ans: Deserialization is the process of reconstructing the object from the serialized state. It is the reverse operation of serialization.

## **What is a transient keyword?**

Ans: If you define any data member as transient, it will not be serialized.

## **What is Externalizable?**

Ans: the Externalizable interface is used to write the state of an object into a byte stream in a compressed format. It is not a marker interface.

## **What is the difference between Serializable and Externalizable interface?**

Ans: Serializable is a marker interface, but Externalizable is not a marker interface. When you use the Serializable interface, your class is serialized automatically by default. But you can override writeObject() and readObject() two methods to control more complex object serialization process. When you use Externalizable interface, you have complete control over your class's serialization process.

## **What is a reflection?**

Ans: Reflection is the process of examining or modifying the runtime behaviour of a class at runtime. It is used in:

IDE (Integrated Development Environment), e.g. Eclipse, MyEclipse, NetBeans. Debugger Test Tools etc.

## **Read: DevOps Interview Questions**

### **Can you access the private method from outside the class?**

Ans: Yes, by changing the runtime behaviour of a class if the class is not secured.

### **What are wrapper classes?**

Ans: Wrapper classes are classes that allow primitive types to be accessed as objects.

### **What is a native method?**

Ans: A native method is a method that is implemented in a language other than Java.

### **What is the purpose of the System class?**

Ans: The purpose of the System class is to provide access to system resources.

### **What comes to mind when someone mentions a shallow copy in Java?**

Ans: Object cloning.

### **What is a singleton class?**

Ans: Singleton class means that at any given time only one instance of the class is present, in one JVM.