

## JAVA Interview Questions

① What do you know about JVM, JRE and JDK?

→ JDK Stands for Java Development Kit. It internally contains JRE and JVM where JRE Stands for Java Runtime Environment and ~~JDK~~ JDK provides all the tools to work with Java language. All the Java development tools - Java.exe and jar file contain in JDK. Also JRE is a software which comes with JDK and also we can also download it separately. JVM helps load and execute the class file and helps to convert byte code to machine level language. JVM contains JIT Compiler and JVM Interpreter.

② Is JRE platform dependant or independent?

The Java ~~Runtime~~ Runtime Environment is platform dependent, meaning that it needs to be installed separately for each operating system. This is because the JRE contains the JVM which is ~~responsible~~ responsible for executing Java Byte Codes.

③ Which is ultimate base class in Java class hierarchy? List the name of methods of it?

→ Object class is the ultimate base class in Java.

In Object class there are total 11 methods. 5 non final methods and 6 final methods.

non final methods are `toString()`,  
`equals()`,  
`hashCode()`,  
`Object clone()`  
`finalize()`,

and all the final methods are →

`getClose()`  
`wait()`  
`wait(long time out, int nanos)`  
`wait (long timeout)`  
`notify()`  
`notifyAll();`

④ Which are the reference types in Java?

→ Which variable of non primitive data type reference of the instance. Hence non primitive type is also called as reference type.

There are 4 non primitive / reference types in Java

- ① Interface
- ② Class
- ③ Enum
- ④ Array

⑤ Explain narrowing and widening?

Narrowing → Process of converting value of variable of wider type into narrower type is called as narrowing. In case of narrowing, explicit type casting is mandatory.

Example → Double to int, int to byte.

Widening → Process of converting value of

of narrower type into wider type is called as widening. In case of widening conversion explicit typecasting is optional.  
Example → Char to int, int to float -  
Short to int.

⑥ How will you print "Hello CDAC" Statement on Screen, without semicolon?

→ we can use if statement and use printf method with System.out in the if statement. ~~as~~ we cannot use println or print because those method don't have return type and in if statement need to return type boolean so printf return type is printstream then we can use condition like == null and its work.

```
if (System.out.printf("HelloCDAC") != null){}
```

}

⑦ Can you write Java application without main function? If yes how?

→ In Java, the main function is the entry point for the program. After compile when Java starts executing the class file JVM is first looking for main method. So without main method JVM failed to convert byte code to machine code. So, without main function we cannot write Java application.

⑧ What will happen if we call main method in static block?

→ If you call the main method in static block, the main method will be executed

twice. This is because static blocks are executed when the class is loaded.

So, static block execute first after that main method is called.

Q) In System.out.println. Explain meaning of every word?

→ System is a final class declared in Java.lang package.

Out is a reference of java.io.  
PrintStream class. It is declared  
as public static final field ~~in~~ inside  
System class. PrintStream is a class  
declared in java.io package. System  
class associate with PrintStream class.  
println, print and printf is a non  
static method of java.io.PrintStream  
class.

Q) How will you pass object to the function by reference?

There is no direct way to pass an  
object to a function by reference in  
Java. Java always passes value in  
the function. ~~in~~ But one way to  
pass an object to a function by reference  
is to use a wrapper class. Another way  
pass object in the function by using  
array. After ~~array~~ Array function  
creating we pass reference and catch  
them by using array in arguments.

⑪ Explain Constructor Chaining? How can we achieve it in C++?

After creating constructor to reuse body of existing constructor we can call constructor from another constructor. It is called as constructor chaining.

To achieve constructor chaining in C++, we can use the member initializer list. The member initializer list is a list of comma-separated expression that are evaluated when an object is constructed.

```
class demo {  
public:  
    demo(int n) : x(n){}
```

⑫ Which are the rules to overload method in sub class?

① The method name must be the same as the method name in the base class.

② The method signature must be different from the method signature in the base class. This means that the number of parameters or the return type must be different.

③ The access modifier of subclass is same as base class modifier or wider than the super class.

(13) Explain the difference among finalize and dispose in Java?

→ Finalize() and Dispose are two methods in Java that are used to clean up ~~the~~ resources that are used by an Object. The main difference between the two methods is the Finalize() is called by the Garbage Collector when an object is no longer needed while Dispose() is called explicitly by the developer. Finalize is protected method and Dispose() is a public method.

(14) Explain the difference among final, finally and finalize?

→ Final is a keyword ~~to~~ for make the field, method or class constant like we can say we cannot change, or override the method, fields and ~~to~~ for final class we cannot create any sub class of final class.

Finally is a block of code that is always executed, regardless of whether an exception is thrown. The finally block is typically used to clean up resources.

Finalize is a protected method that is inherited from object class. It is called by the Garbage Collector when an object is no longer needed.

⑯ Explain the difference among checked and unchecked exception?

→ Java.lang.Exception is considered as super class of all the checked exception. Exception class and its subclasses except RuntimeException are considered as checked exception. Compiler forces developer to handle to use try-catch blocks for checked exception.

Example → Java.lang.Exception  
Java.lang.InterruptedException

All the Java.lang.RuntimeException methods and their subclasses are considered as unchecked exception. Compiler don't force to handle those exception.

Example → RuntimeException  
ArithmaticException  
NullPointerException

⑰ Explain Exception Chaining.

→ Exception chaining is a technique that allows you to associate one exception with another exception. This can be useful for debugging and troubleshooting purpose.

⑱ Explain the difference among throw and throws?

→ Throw is used to explicitly throw an exception from a method or block. The exception that's

thrown must be an instance of a class that inherits from Throwable class.

throws is used to declare that exceptions that a method can potentially throw. That exception handle by compiler itself.

⑯ In which case, finally block doesn't execute?

→ The finally block in Java is a block of code that is always executed regardless of whether an exception is thrown. But if the JVM exits before the finally block then its not executed.

⑰ Explain upcasting?  
process of converting reference of sub class into reference of super class its called upcasting. In subclass all the super class members inherit so we can access them by sub class. Also super class field method gets space into sub class. So by super class reference we can access subclass methods, field which are inherit from super class. It's called upcasting.

⑱ Explain Dynamic method Dispatch  
→ In case of upcasting process of calling subclass method using super class reference it's called Dynamic method Dispatch.

21) What do you know about final methods?

→ A final method in Java is a method that cannot be overridden by subclasses. When we know the method is fully done ~~done~~ with its declaration no need to change then we make it final. It's also should not modified in subclass also.

Example →

```
public final class MyClass  
    public final void Test()  
        System.out.println("Hello")  
    }  
    public final  
    // This method cannot be
```

overridden.

22) Explain fragile base class problem and how can we overcome it?

→ The fragile base class problem is a software ~~design~~ design issue that occurs when a change to a base class causes unexpected behaviors in its subclasses. This can happen because subclasses often rely on the implementation details of their base class. When the base class is changed the subclass may not longer work as expected.

Q23) Why Java does not support multiple implementation inheritance?  
→ Java does not support multiple implementation inheritance because it can lead to the diamond problem.  
The diamond problem is a situation where a class inherits from two base classes that both define the same method. When the subclass calls the method, the compiler is unable to determine which base class method to call. That's why ignoring Diamond problem after C++ multiple inheritance removed from Java.

Q24) Explain marker interface? List the name of some marker interfaces?  
→ A marker interface is an interface in Java that does not have any methods or fields. It is simply a tag that can be used to indicate that a class has certain properties and behaviors. It's used to implement design patterns and frameworks.

Some marker interfaces →

- ① Serializable
- ② Clonable
- ③ Remote
- ④ Autoclosable

25 Explain the significance of marker interface?

→ A marker interface is a Java interface that does not include any methods or fields. It is used to identify classes that share a common characteristic or behavior without affecting their implementation.