Core Java Interview Questions and Answers

MAY 29, 2018 BY PANKAJ

Java interview questions help you in preparing for java based interviews. Whether you are a fresher or highly experienced professional, core java plays a vital role in any Java/JEE interview. Core Java is the favorite area in most of the interviews and plays a crucial role in deciding the outcome of your interview. This post is about **core java interview questions** that comes directly from my 12+ years of Java programming and lots of interviewing experience. Java 10 has been released recently and I have updated the post to include some of the questions from Java 10 and Java 9 releases.

Core Java Interview Questions

I have already written a lot about **java interview questions** for specific topics such as String, Collections and Multithreading.

1. Java String Interview Questions
2. Java Thread Interview Questions
3. Java Collections Interview Questions
4. Java Exception Interview Questions

Here I am providing some of the important core java interview questions with answers that you should know. You can bookmark this post to brush up your knowledge before heading for an interview.

Quizzes are fun, aren't they! I recently published core java quiz of 21 questions. It has been taken by more than 2000 people with an average score of 31.45%. You should take that and try to get your name into the leaderboard.

Here is the link that opens in a new tab: Core Java Quiz

Core Java Interview Questions and Answers

1. What are the important features of Java 10 release?

Java 10 is the first every-six-months from Oracle corporation, so it's not a major release like earlier versions. However some of the important features of Java 10 are:

- Local-Variable Type Inference
- Enhance java.util.Locale and related APIs to implement additional Unicode extensions of BCP 47 language tags.
- Enable the HotSpot VM to allocate the Java object heap on an alternative memory device, such as an NV-DIMM, specified by the user.
- Provide a default set of root Certification Authority (CA) certificates in the JDK.

Java 10 is mostly a maintenance release, however I really liked the local variable type inference feature.

2. What are the important features of Java 9 release?

Java 9 was a major release and brought a lot of features. Some of the important features are:

- Java 9 REPL (JShell)
- Java 9 Module System
- Factory Methods for Immutable List, Set, Map and Map.Entry
- Private methods in Interfaces
- Reactive Streams
- GC (Garbage Collector) Improvements

You will find more details about them at Java 9 Features.

3. What are the important features of Java 8 release?

Java 8 has been released in March 2014, so it's one of the hot topic in java interview questions. If you answer this question clearly, it will show that you like to keep yourself up-to-date with the latest technologies.

Java 8 has been one of the biggest release after Java 5 annotations and generics. Some of the important features of Java 8 are:

1. Interface changes with default and static methods
2. Functional interfaces and Lambda Expressions
3. Java Stream API for collection classes

4. Java Date Time API

I strongly recommend to go through above links to get proper understanding of each one of them, also read Java 8 Features.

4. Name some OOPS Concepts in Java?

Java is based on Object Oriented Programming Concepts, following are some of the OOPS concepts implemented in java programming.

- Abstraction
- Encapsulation
- Polymorphism
- Inheritance
- Association
- Aggregation
- Composition

Read more about them at OOPS Concepts in Java.

5. What do you mean by platform independence of Java?

Platform independence means that you can run the same Java Program in any Operating System. For example, you can write java program in Windows and run it in Mac OS.

6. What is JVM and is it platform independent?

Java Virtual Machine (JVM) is the heart of java programming language. JVM is responsible for converting byte code into machine readable code. JVM is not platform independent, thats why you have different JVM for different operating systems. We can customize JVM with Java Options, such as allocating minimum and maximum memory to JVM. It's called virtual because it provides an interface that doesn't depend on the underlying OS.

7. What is the difference between JDK and JVM?

Java Development Kit (JDK) is for development purpose and JVM is a part of it to execute the java programs.

JDK provides all the tools, executables and binaries required to compile, debug and execute a Java Program. The execution part is handled by JVM to provide machine independence.

8. What is the difference between JVM and JRE?

Java Runtime Environment (JRE) is the implementation of JVM. JRE consists of JVM and java binaries and other classes to execute any program successfully. JRE doesn't contain any development tools like java compiler, debugger etc. If you want to execute any java program, you should have JRE installed.

9. Which class is the superclass of all classes?

`java.lang.Object` is the root class for all the java classes and we don't need to extend it.

10. Why Java doesn't support multiple inheritance?

Java doesn't support multiple inheritance in classes because of "Diamond Problem". To know more about diamond problem with example, read Multiple Inheritance in Java.

However multiple inheritances are supported in interfaces. An interface can extend multiple interfaces because they just declare the methods and implementation will be present in the implementing class.

So there is no issue of the diamond problem with interfaces.

11. Why Java is not pure Object Oriented language?

Java is not said to be pure object-oriented because it supports primitive types such as int, byte, short, long etc. I believe it brings simplicity to the language while writing our code. Obviously, java could have wrapper objects for the primitive types but just for the representation, they would not have provided any benefit.

As we know, for all the primitive types we have wrapper classes such as Integer, Long etc that provides some additional methods.

12. What is difference between path and classpath variables?

PATH is an environment variable used by operating system to locate the executables. That's why when we install Java or want any executable to be found by OS, we need to add the directory location in the PATH variable. If you work on Windows OS, read this post to learn how to setup PATH variable on Windows.

Classpath is specific to java and used by java executables to locate class files. We can provide the classpath location while running java application and it can be a directory, ZIP files, JAR files etc.

13. What is the importance of main method in Java?

main() method is the entry point of any standalone java application. The syntax of main method is `public static void main(String args[])`.

Java main method is public and static so that Java runtime can access it without initializing the class. The input parameter is an array of String through which we can pass runtime arguments to the java program. Check this post to learn how to compile and run java program.

14. What is overloading and overriding in java?

When we have more than one method with the same name in a single class but the arguments are different, then it is called as method overloading.

Overriding concept comes in picture with inheritance when we have two methods with same signature, one in parent class and another in child class. We can use @Override annotation in the child class overridden method to make sure if parent class method is changed, so as child class.

15. Can we overload main method?

Yes, we can have multiple methods with name “main” in a single class. However if we run the class, java runtime environment will look for main method with syntax as `public static void main(String args[])`.

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

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.

17. What is Java Package and which package is imported by default?

Java package is the mechanism to organize the java classes by grouping them. The grouping logic can be based on functionality or modules based. A java class fully classified name contains package and class name. For example, `java.lang.Object` is the fully classified name of `Object` class that is part of `java.lang` package.

`java.lang` package is imported by default and we don't need to import any class from this package explicitly.

18. What are access modifiers?

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

19. What is final keyword?

final keyword is used with Class to make sure no other class can extend it, for example String class is final and we can't extend it.

We can use the final keyword with methods to make sure child classes can't override it.

final keyword can be used with variables to make sure that it can be assigned only once. However the state of the variable can be changed, for example, we can assign a final variable to an object only once but the object variables can change later on.

Java interface variables are by default final and static.

20. What is static keyword?

static keyword can be used with class level variables to make it global i.e all the objects will share the same variable.

static keyword can be used with methods also. A static method can access only static variables of class and invoke only static methods of the class.

Read more in detail at java static keyword.

21. What is finally and finalize in java?

finally block is used with try-catch to put the code that you want to get executed always, even if any exception is thrown by the try-catch block. finally block is mostly used to release resources created in the try block.

finalize() is a special method in Object class that we can override in our classes. This method gets called by the garbage collector when the object is getting garbage collected. This method is usually overridden to release system resources when the object is garbage collected.

22. Can we declare a class as static?

We can't declare a top-level class as static however an inner class can be declared as static. If inner class is declared as static, it's called static nested class.
The static nested class is same as any other top-level class and is nested for only packaging convenience.

Read more about inner classes at java inner class.

23. What is static import?

If we have to use any static variable or method from other class, usually we import the class and then use the method/variable with class name.

```
import java.lang.Math;

//inside class
double test = Math.PI * 5;
```

We can do the same thing by importing the static method or variable only and then use it in the class as if it belongs to it.

```
import static java.lang.Math.PI;

//no need to refer class now
double test = PI * 5;
```

Use of static import can cause confusion, so it's better to avoid it. Overuse of static import can make your program unreadable and unmaintainable.

24. What is try-with-resources in java?

One of the Java 7 features is the try-with-resources statement for automatic resource management. Before Java 7, there was no auto resource management and we should explicitly close the resource. Usually, it was done in the `finally` block of a try-catch statement. This approach used to cause memory leaks when we forgot to close the resource.

From Java 7, we can create resources inside try block and use it. Java takes care of closing it as soon as try-catch block gets finished. Read more at Java Automatic Resource Management.

25. What is multi-catch block in java?

Java 7 one of the improvement was multi-catch block where we can catch multiple exceptions in a single catch block. This makes are code shorter and cleaner when every catch block has similar code.

If a catch block handles multiple exceptions, you can separate them using a pipe (|) and in this case, exception parameter (ex) is final, so you can't change it.

Read more at Java multi catch block.

26. What is static block?

Java static block is the group of statements that gets executed when the class is loaded into memory by Java ClassLoader. It is used to initialize static variables of the class. Mostly it's used to create static resources when class is loaded.

27. What is an interface?

Interfaces are core part of java programming language and used a lot not only in JDK but also java design patterns, most of the frameworks and tools. Interfaces provide a way to achieve abstraction in java and used to define the contract for the subclasses to implement.

Interfaces are good for starting point to define Type and create top level hierarchy in our code. Since a java class can implements multiple interfaces, it's better to use interfaces as super class in most of the cases. Read more at java interface.

28. What is an abstract class?

Abstract classes are used in java to create a class with some default method implementation for subclasses. An abstract class can have abstract method without body and it can have methods with implementation also.

abstract keyword is used to create a abstract class. Abstract classes can't be instantiated and mostly used to provide base for sub-classes to extend and implement the abstract methods and override or use the implemented methods in abstract class. Read important points about abstract classes at java abstract class.

29. What is the difference between abstract class and interface?

abstract keyword is used to create abstract class whereas interface is the keyword for interfaces.

Abstract classes can have method implementations whereas interfaces can't.

A class can extend only one abstract class but it can implement multiple interfaces.

We can run abstract class if it has main() method whereas we can't run an interface.

Some more differences in detail are at Difference between Abstract Class and Interface.

30. Can an interface implement or extend another interface?

Interfaces don't implement another interface, they extend it. Since interfaces can't have method implementations, there is no issue of diamond problem. That's why we have multiple inheritance in interfaces i.e an interface can extend multiple interfaces.

From Java 8 onwards, interfaces can have default method implementations. So to handle diamond problem when a common default method is present in multiple interfaces, it's mandatory to provide implementation of the method in the class implementing them. For more details with examples, read Java 8 interface changes.

31. What is Marker interface?

A marker interface is an empty interface without any method but used to force some functionality in implementing classes by Java. Some of the well known marker interfaces are Serializable and Cloneable.

32. What are Wrapper classes?

Java wrapper classes are the Object representation of eight primitive types in java. All the wrapper classes in java are immutable and final. Java 5 autoboxing and unboxing allows easy conversion between primitive types and their corresponding wrapper classes.

Read more at Wrapper classes in Java.

33. What is Enum in Java?

Enum was introduced in Java 1.5 as a new type whose fields consists of fixed set of constants. For example, in Java we can create Direction as enum with fixed fields as EAST, WEST, NORTH, SOUTH.

enum is the keyword to create an enum type and similar to the class. Enum constants are implicitly static and final. Read more in detail at java enum.

34. What is Java Annotations?

Java Annotations provide information about the code and they have no direct effect on the code they annotate. Annotations are introduced in Java 5. Annotation is metadata about the program embedded in the program itself. It can be parsed by the annotation parsing tool or by the compiler. We can also specify annotation availability to either compile time only or till runtime also. Java Built-in annotations are @Override, @Deprecated and @SuppressWarnings. Read more at java annotations.

35. What is Java Reflection API? Why it's so important to have?

Java Reflection API provides the ability to inspect and modify the runtime behavior of java application. We can inspect a java class, interface, enum and get their methods and field details. Reflection API is an advanced topic and we should avoid it in normal programming. Reflection API usage can break the design pattern such as Singleton pattern by invoking the private constructor i.e violating the rules of access modifiers.

Even though we don't use Reflection API in normal programming, it's very important to have. We can't have any frameworks such as Spring, Hibernate or servers such as Tomcat, JBoss without Reflection API. They invoke the appropriate methods and instantiate classes through reflection API and use it a lot for other processing.

Read Java Reflection Tutorial to get in-depth knowledge of reflection api.

36. What is composition in java?

Composition is the design technique to implement has-a relationship in classes. We can use Object composition for code reuse.

Java composition is achieved by using instance variables that refer to other objects. The benefit of using composition is that we can control the visibility of other objects to client classes and reuse only what we need. Read more with example at Java Composition example.

37. What is the benefit of Composition over Inheritance?

One of the best practices of Java programming is to "favor composition over inheritance". Some of the possible reasons are:

- Any change in the superclass might affect subclass even though we might not be using the superclass methods. For example, if we have a method test() in the subclass and suddenly somebody introduces a method test() in the superclass, we will get compilation errors in the subclass. Composition will never face this issue because we are using only what methods we need.
- Inheritance exposes all the superclass methods and variables to the client and if we have no control in designing superclass, it can lead to security holes. Composition allows us to provide restricted access to the methods and hence more secure.
- We can get runtime binding in composition where inheritance binds the classes at compile time. So composition provides flexibility in the invocation of methods.

You can read more about above benefits of composition over inheritance at java composition vs inheritance.

38. How to sort a collection of custom Objects in Java?

We need to implement Comparable interface to support sorting of custom objects in a collection. Comparable interface has compareTo(T obj) method which is used by sorting methods and by providing this method implementation, we can provide default way to sort custom objects collection.

However, if you want to sort based on different criteria, such as sorting an Employees collection based on salary or age, then we can create Comparator instances and pass it as sorting methodology. For more details read Java Comparable and Comparator.

39. What is inner class in java?

We can define a class inside a class and they are called nested classes. Any non-static nested class is known as inner class. Inner classes are associated with the object of the class and they can access all the variables and methods of the outer class. Since inner classes are associated with the instance, we can't have any static variables in them.

We can have local inner class or anonymous inner class inside a class. For more details read java inner class.

40. What is anonymous inner class?

A local inner class without name is known as anonymous inner class. An anonymous class is defined and instantiated in a single statement. Anonymous inner class always extend a class or implement an interface.

Since an anonymous class has no name, it is not possible to define a constructor for an anonymous class. Anonymous inner classes are accessible only at the point where it is defined.

41. What is Classloader in Java?

Java Classloader is the program that loads byte code program into memory when we want to access any class. We can create our own classloader by extending ClassLoader class and overriding loadClass(String name) method. Learn more at java classloader.

42. What are different types of classloaders?

There are three types of built-in Class Loaders in Java:

1. Bootstrap Class Loader – It loads JDK internal classes, typically loads rt.jar and other core classes.
2. Extensions Class Loader – It loads classes from the JDK extensions directory, usually $JAVA_HOME/lib/ext directory.

3. System Class Loader – It loads classes from the current classpath that can be set while invoking a program using -cp or -classpath command line options.

43. What is ternary operator in java?

Java ternary operator is the only conditional operator that takes three operands. It's a one liner replacement for if-then-else statement and used a lot in java programming. We can use ternary operator if-else conditions or even switch conditions using nested ternary operators. An example can be found at java ternary operator.

44. What does super keyword do?

super keyword can be used to access super class method when you have overridden the method in the child class.

We can use super keyword to invoke superclass constructor in child class constructor but in this case, it should be the first statement in the constructor method.

```
package com.journaldev.access;

public class SuperClass {

        public SuperClass(){
        }

        public SuperClass(int i){}

        public void test(){
                System.out.println("super class test method");
        }
}
```

Use of super keyword can be seen in below child class implementation.

```
package com.journaldev.access;

public class ChildClass extends SuperClass {

        public ChildClass(String str){
                //access super class constructor with super keyword
                super();

                //access child class method
                test();

                //use super to access super class method
                super.test();
        }

        @Override
        public void test(){
                System.out.println("child class test method");
        }
}
```

45. What is break and continue statement?

We can use break statement to terminate for, while, or do-while loop. We can use break statement in switch statement to exit the switch case. You can see the example of break statement at java break. We can use break with label to terminate the nested loops.

The continue statement skips the current iteration of a for, while or do-while loop. We can use continue statement with the label to skip the current iteration of the outermost loop.

46. What is this keyword?

this keyword provides the reference to the current object and it's mostly used to make sure that object variables are used, not the local variables having the same name.

```
//constructor
public Point(int x, int y) {
    this.x = x;
    this.y = y;
}
```

We can also use this keyword to invoke other constructors from a constructor.

```
public Rectangle() {
    this(0, 0, 0, 0);
}
public Rectangle(int width, int height) {
    this(0, 0, width, height);
}
public Rectangle(int x, int y, int width, int height) {
    this.x = x;
```

```
        this.y = y;
        this.width = width;
        this.height = height;
    }
```

47. What is default constructor?

No argument constructor of a class is known as default constructor. When we don't define any constructor for the class, java compiler automatically creates the default no-args constructor for the class. If there are other constructors defined, then compiler won't create default constructor for us.

48. Can we have try without catch block?

Yes, we can have try-finally statement and hence avoiding catch block.

49. What is Garbage Collection?

Garbage Collection is the process of looking at heap memory, identifying which objects are in use and which are not, and deleting the unused objects. In Java, process of deallocating memory is handled automatically by the garbage collector.

We can run the garbage collector with code `Runtime.getRuntime().gc()` or use utility method `System.gc()`. For a detailed analysis of Heap Memory and Garbage Collection, please read Java Garbage Collection.

50. What is Serialization and Deserialization?

We can convert a Java object to an Stream that is called Serialization. Once an object is converted to Stream, it can be saved to file or send over the network or used in socket connections.

The object should implement Serializable interface and we can use java.io.ObjectOutputStream to write object to file or to any OutputStream object. Read more at Java Serialization.

The process of converting stream data created through serialization to Object is called deserialization. Read more at Java Deserialization.

51. How to run a JAR file through command prompt?

We can run a jar file using java command but it requires Main-Class entry in jar manifest file. Main-Class is the entry point of the jar and used by java command to execute the class. Learn more at java jar file.

52. What is the use of System class?

Java System Class is one of the core classes. One of the easiest way to log information for debugging is System.out.print() method.

System class is final so that we can't subclass and override its behavior through inheritance. System class doesn't provide any public constructors, so we can't instantiate this class and that's why all of its methods are static.

Some of the utility methods of System class are for array copy, get the current time, reading environment variables. Read more at Java System Class.

53. What is instanceof keyword?

We can use instanceof keyword to check if an object belongs to a class or not. We should avoid it's usage as much as possible. Sample usage is:

```
public static void main(String args[]){
        Object str = new String("abc");

        if(str instanceof String){
                System.out.println("String value:"+str);
        }

        if(str instanceof Integer){
                System.out.println("Integer value:"+str);
        }
}
```

Since str is of type String at runtime, first if statement evaluates to true and second one to false.

54. Can we use String with switch case?

One of the Java 7 feature was improvement of switch case of allow Strings. So if you are using Java 7 or higher version, you can use String in switch-case statements. Read more at Java switch-case String example.

55. Java is Pass by Value or Pass by Reference?

This is a very confusing question, we know that object variables contain the reference to the Objects in heap space. When we invoke any method, a copy of these variables is passed and gets stored in the stack memory of the method. We can test any language whether it's pass by reference or pass by value through a simple generic swap method, to learn more read Java is Pass by Value and Not Pass by Reference.

56. What is difference between Heap and Stack Memory?

Major difference between Heap and Stack memory are as follows:

- Heap memory is used by all the parts of the application whereas stack memory is used only by one thread of execution.
- Whenever an object is created, it's always stored in the Heap space and stack memory contains the reference to it. Stack memory only contains local primitive variables and reference variables to objects in heap space.
- Memory management in the stack is done in LIFO manner whereas it's more complex in Heap memory because it's used globally.

For a detailed explanation with a sample program, read Java Heap vs Stack Memory.

57. Java Compiler is stored in JDK, JRE or JVM?

The task of java compiler is to convert java program into bytecode, we have `javac` executable for that. So it must be stored in JDK, we don't need it in JRE and JVM is just the specs.

58. What will be the output of following programs?

1. **static method in class**

```
package com.journaldev.util;

public class Test {

        public static String toString(){
                System.out.println("Test toString called");
                return "";
        }
        public static void main(String args[]){
                System.out.println(toString());
        }
}
```

Answer: The code won't compile because we can't have an `Object` class method with static keyword. Note that Object class has `toString()` method. You will get compile time error as "This static method cannot hide the instance method from Object". The reason is that static method belongs to the class and since every class base is Object, we can't have the same method in the instance as well as in class. You won't get this error if you change the method name from toString() to something else that is not present in superclass `Object`.

2. **static method invocation**

```
package com.journaldev.util;

public class Test {

        public static String foo(){
                System.out.println("Test foo called");
                return "";
        }

        public static void main(String args[]){
                Test obj = null;
                System.out.println(obj.foo());
        }
}
```

Answer: Well this is a strange situation. We all have seen `NullPointerException` when we invoke a method on the object that is NULL. But here this program will work and prints "Test foo called".

The reason for this is the java compiler code optimization. When the java code is compiled to produced byte code, it figures out that foo() is a static method and should be called using class. So it changes the method call `obj.foo()` to `Test.foo()` and hence no `NullPointerException`.

I must admit that it's a very tricky question and if you are interviewing someone, this will blow his mind off. □

I will keep on adding more questions to the list, if you think I missed any important one, please let me know through comments. [no_toc]

FILED UNDER: INTERVIEW QUESTIONS

About Pankaj

If you have come this far, it means that you liked what you are reading. Why not reach little more and connect with me directly on **Google Plus**, **Facebook** or **Twitter**. I would love to hear your thoughts and opinions on my articles directly.

Recently I started creating video tutorials too, so do check out my videos on **Youtube**.

« Struts2 Interview Questions and Answers JDBC Interview Questions and Answers »

Comments

David says

MARCH 19, 2018 AT 11:55 AM

Amazing questions and very good explanation, appreciate the effort you have put for this.

john says

MARCH 8, 2018 AT 3:47 PM

I have been using java for 8+ years commercially longer for my own projects and even longer if considering uni was heavily java oriented.
– I covered all basics got high grades, got excellent results in assignments and solved many problems during
working life.
Yet recently I was asked a basic question java question that I got wrong 'theoretically'.
But by chance I happened to Practically implemented it the nigh before. This dichotomy got me to thinking!
The point I'm making is I've forgotten so much i've done over the years and I don't care because I research it when I need to use it, I'm not ashamed to say I cut and paste a lot from web. Or that I need to revise something to use it.
Its not that I don't know but that I choose to forget and move onto new things, more things and greater variety and depth of aspects.

It is this progress that feeds back into what I learned previously, informs my earlier knowledge and enhances my revision of the core ideas when I need to reutilise them again. This is the secret to keeping knowledge truly vital alive and in use. The same old ideas acquire a new context!
Just understanding them and memorising them over and over makes you just a circus performer.
But sadly most/all companies require a metric for the sake of a metric so evaluations are created for the sake of evaluations.
I'm proud that I have developed over the years my own independence of mind that allows me to grow beyond limits of rhetoric that confounds a more narrow limited point of view.
-A better question to ask is how can I use the knowledge to create something with a practical real use that is new or superior. This is the true test of ability.

Ram Sharan says

FEBRUARY 26, 2018 AT 10:42 AM

I'm extremely thankful to you Mr. Pankaj. You taught me core Java concepts in your way and It's amazing way of teaching. Cannot thank you enough. I would say to add more and more questions and answers here. Always stay motivated to write posts to assist people like us.

Shivi says

FEBRUARY 23, 2018 AT 4:32 AM

It is very useful… very nice questions thanks for sharing with us.. please add some more core java question to it…

pc says

FEBRUARY 13, 2018 AT 1:48 AM

In this you will get latest java Interview Questions ……and many more contents to come

Ankit rai says

FEBRUARY 2, 2018 AT 5:21 AM

In Ans 50)

"(str instanceof Integer)" statement will give compilation issue not false.

For instanceof operator both the operands must have some relation.

sajjan says

FEBRUARY 10, 2018 AT 10:55 PM

Yes …correct.

chetan says

FEBRUARY 15, 2018 AT 10:15 AM

it does not give any error it shows output as

String value:abc

Arul says

FEBRUARY 1, 2018 AT 1:48 PM

Very concise answers to the point. Keep up the good work.

Akash Aher says

JANUARY 29, 2018 AT 9:55 PM

Thank you for helping us to understand java…

Helps a lot

gam3r-z says

JANUARY 29, 2018 AT 4:27 AM

very useful! I l most shocked by the last two responses

keep on

regards

M.Pradeep says

JANUARY 7, 2018 AT 9:41 PM

Thank you so much sir

Prashant Kumar says

DECEMBER 26, 2017 AT 5:37 AM

Hello Pankaj,

Nice article to revise core java.

Please add questions about java collection package also.

Thank you,

Nati Getch says

NOVEMBER 25, 2017 AT 9:32 PM

This article really helped me to revise concepts in core Java. Highly appreciated.

Is the statement correct on question 27? Java 8 has enabled method implementation for interface using default keyword. And I think Diamond problem can happen if a given interface extends two interfaces which have the same default methods.

Pankaj says

NOVEMBER 26, 2017 AT 2:49 PM

Thanks for pointing it out, the answer had to be updated for Java 8 changes. I have updated it, java has taken care of this situation by making it mandatory for class to provide implementation of common default methods in the interfaces it's implementing.

Kapil Bhagia says

NOVEMBER 19, 2017 AT 8:28 AM

Hi Pankaj, please review this android app if you get some time:
https://play.google.com/store/apps/details?id=com.bethexsoftware.javaranking&hl=en
I have developed this app and tried to gamify learning. Please let me know your thoughts.
I was checking your post and I like the comprehensive material you have here. Do you mind if I start making questions for my app using your content?

Lily says

NOVEMBER 14, 2017 AT 8:28 AM

great stuff! It really helped in the Java interview!

renuka says

NOVEMBER 5, 2017 AT 10:40 PM

Great stuff, just great! This Q & A helped every job seeker as well as freshers. Many important questions are covered here.

JanBask Training says

NOVEMBER 2, 2017 AT 11:41 AM

Thanks for sharing

Arul says

OCTOBER 21, 2017 AT 7:49 PM

Pankaj,

Very impressed with your core Java interview questions and answers post with concise to the point answers. Very handy to brush up. Keep up the good work.

Kamal Chaturvedi says

AUGUST 27, 2017 AT 6:11 AM

Great work man .. Learnt a lot today □

Anurag Singh says

JULY 24, 2017 AT 3:10 AM

It is very helpful for me to crack the core java interviews.

keep posting.

thanks

admin says

JULY 18, 2017 AT 1:46 AM

Please make a modification in JAVA 8 Release date.

Anupam Jain says

JULY 8, 2017 AT 9:50 AM

Hi,

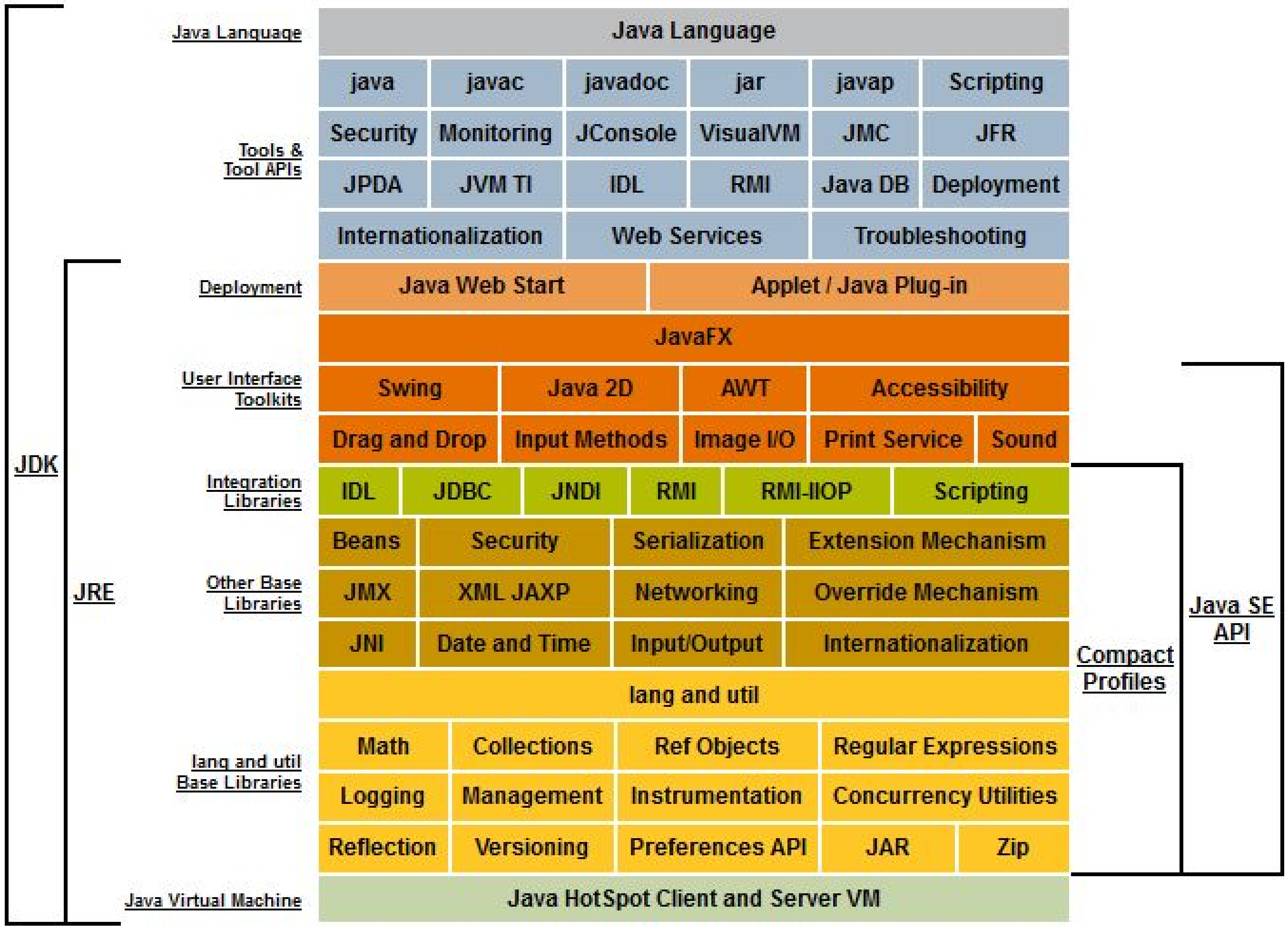
I have a doubt on question 5. Where you've said that JRE consists of JVM and other binaries etc. In my opinion, this is incorrect to say that JRE consists of JVM for two reasons:

1. JVM is more of a specification and JRE is implementation of that.
2. JVM is an abstract thing which would have no physical existence. In fact JVM does not have any knowledge of Java language per se, All it understands is the byte code of .class files.

Please suggest what you think on this?

Pankaj says

JULY 9, 2017 AT 5:17 AM



Java Language: Java Language

Tools & Tool APIs:
java | javac | javadoc | jar | javap | Scripting
Security | Monitoring | JConsole | VisualVM | JMC | JFR
JPDA | JVM TI | IDL | RMI | Java DB | Deployment
Internationalization | Web Services | Troubleshooting

Deployment: Java Web Start | Applet / Java Plug-in

User Interface Toolkits:
JavaFX
Swing | Java 2D | AWT | Accessibility
Drag and Drop | Input Methods | Image I/O | Print Service | Sound

Integration Libraries: IDL | JDBC | JNDI | RMI | RMI-IIOP | Scripting

Other Base Libraries:
Beans | Security | Serialization | Extension Mechanism
JMX | XML JAXP | Networking | Override Mechanism
JNI | Date and Time | Input/Output | Internationalization

lang and util Base Libraries:
lang and util
Math | Collections | Ref Objects | Regular Expressions
Logging | Management | Instrumentation | Concurrency Utilities
Reflection | Versioning | Preferences API | JAR | Zip

Java Virtual Machine: Java HotSpot Client and Server VM

Fresher walkin says

MAY 1, 2017 AT 9:30 AM

Find more IT/Software Walk in Jobs in India. Get hired by TOP MNN's .

Priyanka Chordia says

DECEMBER 21, 2016 AT 2:34 AM

Thank You Pankaj for sharing this with us.It is very helpful for core java interview.

SamJozi says

SEPTEMBER 23, 2016 AT 7:01 PM

Hi Pankaj

Thank you for your great site and wonderful explanation. I am a newbie and benefited a lot from your site.

I have real difficulty understanding the last question. I am able to use static object methods in my own programs and no exception returned.

I even copy paste your program to my own IDE and saw the error but after changing the method name ,the error disappeared..

and for the second part of your last question you have called static object methods there. I really need any help here. any explanation would be much appreciated.

Pankaj says

SEPTEMBER 23, 2016 AT 10:39 PM

Hi Sam,

I have updated the explanation of questions. Note that toString() method is present in Object class and hence the error. Since you changed the name of method, this conflict is not there and no error.

For second part, explanation is very clear. Because of compiler optimization of byte code, we are not getting exception since it changes from object to Class reference while calling static method.

SamJozi says

SEPTEMBER 24, 2016 AT 9:40 PM

Thanks Pankaj for your explanation , now it is more clear. You might want to add the the fact at the fact that static methods can not be overridden but can be hidden and the reason for that.

Nishanth says

AUGUST 25, 2016 AT 7:45 PM

Hi Pankaj,
I truly appreciate your efforts in writing the answers.Please look into Question-28.Provided link is not working.

Nishanth says

AUGUST 25, 2016 AT 7:46 PM

Sorry it's question no-26.

Pankaj says

AUGUST 26, 2016 AT 3:08 AM

Thanks for pointing it out, I have corrected the link.

Rashid says

JULY 17, 2016 AT 11:24 AM

Part of collection question is missing.

Pankaj says

JULY 17, 2016 AT 9:41 PM

what do you mean?

Amit Thumar says

JULY 1, 2016 AT 9:55 AM

Very helpful to get all concept brushed up on a single page. Last one question was mind-blowing! Thanks!

Java Training Institutes in Bangalore says

JUNE 21, 2016 AT 12:09 AM

This is very very helpful post for people like me for preparing interview. Also, very good and descriptive answer for all questions.

Shravani says

MAY 5, 2016 AT 8:07 AM

Thanks a lot!!!!! Very useful questions!

Prasanna says

APRIL 27, 2016 AT 4:00 AM

Hi Pankaj,
In one of my recent interview I got a question how java is able to identify errors while compilation with mentioning line numbers of error. My answer was Java compiler is capable of finding such errors while compiling. But still interviewer is not convenient. Can you help me to answer this in better way and let me know how compiler internally works to identify syntax or semantics errors?

> **Prasanna says**
>
> MAY 13, 2016 AT 1:41 AM
>
> Pankaj,
> Can you answer or someone my question?

qiangyu says

MARCH 6, 2016 AT 10:14 PM

you are so cool

vikas rathore says

FEBRUARY 18, 2016 AT 2:11 AM

static keword use five way ,
which one executed first static variable,static block,static method,ineer static block,interface static method.
please tell me their order .
iam confused abt it,
correct me.

Riley says

JANUARY 9, 2016 AT 7:51 AM

Hi again Pankaj,
great list – one thing, Java8 changes some of the answers, comparing abstract classes to interfaces, for one, as now interfaces may contain (default & static) methods
thanks for your great site!

senthil.narayanan2013@gmail.com says

DECEMBER 3, 2015 AT 2:58 PM

Thanks you very much pankaj…
I referred many link but this one very nice explanation & excellent stuff. We expecting similar stuff like Web Service (SOAP/REST) as well.

-Regards
Senthil

Pankaj says

DECEMBER 3, 2015 AT 10:07 PM

Web services are also covered in detail, please check below link.
https://www.journaldev.com/9193/web-services-interview-questions-soap-and-rest
https://www.journaldev.com/dev/java/j2ee

senthil.narayanan2013@gmail.com says

DECEMBER 4, 2015 AT 12:36 PM

Excellent, Thanks a lot pankaj

Babu Vayila says

JULY 6, 2016 AT 9:28 PM

Hi pankaj,
Can I expect any PDF documentation so that I can download and go through all the topics when I have free time.

Anitha says

OCTOBER 18, 2015 AT 10:44 PM

Tell me the difference between encapsulation and abstraction , where those concepts are implemented.

Peter Clark says

SEPTEMBER 28, 2015 AT 4:43 AM

Thanks for sharing a java serialization question and answer, It's really helpful and I would like to share latest java and java script interview question and answer for freshers and experienced.

Mustafa says

SEPTEMBER 22, 2015 AT 7:02 AM

For #26 Difference between Abstract & Interface. Abstract classes can't be instantiated so how can you run it from main?

jayachand Mannam says

NOVEMBER 21, 2015 AT 5:55 PM

You have to use subclass of the Abstract class that implements all the abstract methods.

srinivas says

NOVEMBER 22, 2015 AT 11:23 PM

We can initiate the object for the Abstract class but we can run the class with main method.

```
public abstract class TestAbstract {
public abstract void sayHi();
public void printHi(){
System.out.println(“Hello”);
}
public static void main(String[] args) {
System.out.println(“IN the Main”);
}
}
```

indrajeet yadav says

JULY 17, 2015 AT 5:47 PM

hi Pankaj Sir,

i have small doubt in java, new is a keyword or new is an operator. If key then my next question is then what is exact difference between keyword and operator and when we can say this is an operator or this is a keyword?

Praveenkumar kodge says

AUGUST 12, 2015 AT 1:10 AM

answer :

hello

new is keyword it’s not an operator,

keyword has predefined meaning
operator helps u for doing an operation like
arithmetic, assignment using +,= we can perform

ramu says

JULY 27, 2017 AT 11:41 PM

new is used to allocate memory , operator is used to perform operations like arithmetic ,relation,boolean expressions.

Chiranjeevi says

MAY 9, 2015 AT 8:39 AM

excellent sir

Hanna says

APRIL 23, 2015 AT 12:33 PM

Thanks for the article.
Minor correction to #46:
Garbage Collector performs automatic memory management. So it's responsible not only for recovering memory used by objects that are no longer reachable but also for allocating memory.
See Java Memory Management specification for details.

andy says

APRIL 2, 2015 AT 3:42 AM

the best ever collection of java questions and answers

lv85660 says

MARCH 26, 2015 AT 7:59 PM

Good Stuff.Really helpful.

vivek says

MARCH 22, 2015 AT 2:04 PM

hey link is broken on question number 26 – Difference between Abstract Class and Interface.

href is having a typo.

Suneesh says

MARCH 18, 2015 AT 7:41 AM

Hi , Thanks a lot..

and tel me wat is pre- processor and post processor in java

what are the condition to override equals and hashCode in a class

we are not using Pointer in java , but smtnk is tr to replace it, wat s tat

Somanath Behera says

FEBRUARY 7, 2015 AT 6:42 PM

Dear Sir,

Hands off to you. This is really helpful for any java professionals.

Supriya says

DECEMBER 23, 2014 AT 12:09 AM

Thanks for helping..

Pabel Lopez says

NOVEMBER 21, 2014 AT 5:01 PM

I would be great to add also the new features of java 7, even if is not the latest version it was some relevant adds compared with java 6.

Thank you for all the time dedicated to your site.

Pabel Lopez says

NOVEMBER 24, 2014 AT 7:01 PM

Question 26 is missing the hyperlink.

Siva says

SEPTEMBER 20, 2014 AT 4:02 AM

Hi Pankaj,
Since we're spending maximum hours in offices, we're accessing your website during office hours maximum. So we're unable to read full questions since your website asking to like on "Facebook, Twitter or Google Plus". But all these three are banned sites in offices. So we're unable to read all the questions & answers in office. But in Home we're able to access freely, since we already gave like for your page in FB. We would be very great full, if you somehow find a solution for this problem. — Thanks in advance □

abhinav says

NOVEMBER 14, 2014 AT 12:12 PM

Same problem here too………

□

Robert says

JANUARY 12, 2015 AT 9:58 AM

Close your browser, clear your browser cache and start again. You will have access to all questions.

Sorrowfull Blinger says

AUGUST 28, 2014 AT 10:32 PM

Much Appreciated … great set of questions ..please keep this as updated as possible….

Ravi says

AUGUST 28, 2014 AT 6:01 AM

In Inheritance if you create object for only sub class,is super class object is created or not?

mahi says

AUGUST 8, 2014 AT 6:36 PM

hai sir,

where is all questions answers please inform i will wait your answer?

Vignesh says

AUGUST 7, 2014 AT 1:29 AM

For the 52 question A,

Executing the code which you have written dint throw any error. Since we're accessing the static method call.

and also for the question B, i havn't got the Nullpointerexception.

Both the error occured while removing the static keyword from the method toString.

Please look at the issue

Pankaj says

AUGUST 7, 2014 AT 2:22 AM

No these are correct, please check your code. You must be doing some mistake.

Vignesh says

AUGUST 7, 2014 AT 3:59 AM

You're right i was actually doing a mistake. Instead of toString i was using someother method name. Thanks for the good article.

Azhaguvel A says

JULY 18, 2014 AT 11:03 AM

Thank you so much for your efforts □

Ammu says

JULY 7, 2014 AT 9:19 PM

I m really happy by seeing this site just by refering this site we can glance up with java concepts for interview

Thank You sooo much,,,,,,,

Swapnil Kadu says

JULY 5, 2014 AT 11:19 PM

Integer c = new Integer(5);

Integer d = c;

Integer e = 5;

int f = 5;

System.out.println(c==d); // returns true ..

System.out.println(c==e); // returns false ..

System.out.println(c==f); // returns true .. ??

System.out.println(e==f); // returns true .. ??

Can u explain this with memory allocation …

Thank You

Silviu says

AUGUST 28, 2014 AT 5:04 AM

"The JVM is caching Integer values. == only works for numbers between -128 and 127"

http://stackoverflow.com/questions/3637936/java-integer-equals-vs

Gupta Ji says

JULY 16, 2016 AT 10:44 PM

Still same answer is coming when i changed value from 5 to 111115 , But as per silviu, The JVM is caching Integer values. == only works for numbers between -128 and 127" ?? not able to understand, can someone plz. help?

```
Integer c = new Integer(111115);
Integer d = c;
Integer e = 111115;
int f = 111115;
System.out.println(c==d); // returns true ..
System.out.println(c==e); // returns false ..
System.out.println(c==f); // returns true .. ??
System.out.println(e==f); // returns true .. ??
```

Pankaj says

JULY 17, 2016 AT 6:03 AM

when comparing with f, Integer objects int value will be used. Since int is primitive data type, the value is used for comparison and hence TRUE in last two statements.

Silviu says

AUGUST 28, 2014 AT 5:12 AM

within Java 7, depends on IntegerCache

Silviu says

SEPTEMBER 22, 2014 AT 1:32 PM

but, in your example, it's all about unboxing (Integer to int)

prasanth says

JULY 1, 2014 AT 5:23 AM

It is being very useful site please keep updating us with all the database and languages in trend THANK YOU Very much

Prasanthi says

JUNE 21, 2014 AT 1:28 AM

Very nice.And it will be very useful.

Ashish says

JUNE 19, 2014 AT 7:51 PM

Awesome Post Pankaj……..very informative

Thanks

Manjunath Prabhu says

JUNE 11, 2014 AT 2:17 AM

Excellent … very helpful…. Thanks a lot Pankaj □ □

Abhimanyu says

JUNE 11, 2014 AT 12:18 AM

This was really nice □

Madhusmita Nayak says

JUNE 6, 2014 AT 11:50 PM

Can we override main method?

Dilip singh says

JUNE 23, 2014 AT 3:27 AM

No , You can not override static methods.

Raghava says

JUNE 3, 2014 AT 5:38 PM

Really impressive .. i had been to various sites for reading java/jee,
i haven't find in any of them are at least near as good as this site…
finally i find this with neat, nice explanation and navigation of links when necessary …
even i read most of the comments who are anticipated with writer… and i understood those comments are not made with clear understanding (Ex: Java Tricky Programming Questions — comments)
i read here java associated topics, Questions etc… i felt it is ultimate…
I am really thankful to you for providing such great stuff.
Expecting similar stuff w.r.t Hibernate as well.
Thanks
Raghava

Pankaj says

JUNE 3, 2014 AT 8:02 PM

Thanks Raghava, Hibernate interview questions post will be coming soon, when I am done with it's tutorial series.

sachin anthwal says

MAY 18, 2014 AT 3:24 AM

Hi,
Can you please tell me why should we override equals and hascode method in a code?

Pankaj says

MAY 18, 2014 AT 12:42 PM

That's a very important question and very closely goes with Collections, that's why it's there in the Collections interview questions.
Please go through questions 18,19,20 from https://www.journaldev.com/1330/java-collections-interview-questions-and-answers#hashcode-equals

sachin anthwal says

MAY 18, 2014 AT 3:22 AM

Hi,
I think this is a basic interview questions which you have listed above but I hope these all have to be in different way for experienced guys. I give you one example over here so that you can assume my thought as mentioned below
class abc extends Exception
{
}
if we use or throw abc class so that will be considered as a checked/unchecked exception if yes so why?
Thanks

Pankaj says

MAY 18, 2014 AT 12:36 PM

This is a basic question, since Exception is a checked exception… all of it's subclasses will be too.
For more exception related interview questions, please read
https://www.journaldev.com/2167/java-exception-interview-questions-and-answers

HungryToLearn says

MAY 18, 2014 AT 2:27 AM

Hi Pankaj,

Hats off…. Its a very nice material which helps a lot.

I am unable to see answers from 15th to 52nd question. Could you please help me on that

Pankaj says

MAY 18, 2014 AT 2:45 AM

you need to click on one of the social buttons to unlock them.

Aftab Ahmed says

MAY 22, 2014 AT 10:42 AM

Not able to unlock questions. Social buttons not working.

Pankaj says

MAY 22, 2014 AT 8:30 PM

It works on Google Chrome, Firefox, Safari and all the leading browsers. I have tested it myself.

HungryToLearn says

MAY 18, 2014 AT 5:52 AM

Unable to ope the link “Difference between Abstract Class and Interface” in 26th Question

santosh says

MAY 15, 2014 AT 10:01 PM

Very hrlpful site.thanks a lot pankaj

it will be really hrlpful if u cn post info and faqs on webservices(restful)

Pankaj says

MAY 16, 2014 AT 12:56 AM

Okay, I will write something about them in some time.

santosh says

MAY 15, 2014 AT 9:59 PM

Very good site with lots of usefull info to prepare for interviews

pk says

APRIL 25, 2014 AT 3:53 AM

gud one….

Nagendra says

APRIL 18, 2014 AT 10:06 PM

This is site is best place to prepare interviews without opening any book. Here JVM Architecture is missing any plans to write on JVM Architecture

Ravikumar says

APRIL 10, 2014 AT 3:06 PM

simply supurb..

Ravikumar says

APRIL 10, 2014 AT 3:05 PM

simply supurb.. You can try to post all like (J2ee, Framework, Database also..) It was great useful for us… Thanks…

erikala reddy says

MARCH 31, 2014 AT 4:07 PM

Thank you very much for selection of good questions.

ravi says

MARCH 30, 2014 AT 6:19 PM

In question no 48 u specified that …System class is final and static ……Is it System class static ?

> **Pankaj says**
>
> MARCH 30, 2014 AT 7:40 PM
>
> thanks for catching it, obviously a class can't be static. Corrected the typo error.

Anita says

MARCH 27, 2014 AT 1:52 AM

Great article! Very good selection of questions and short and clear answers. Loved it! Thanks for sharing!!!

Anand Kumar says

MARCH 21, 2014 AT 1:54 PM

Thanks a lot Pankaj. It helped me to much in quick review of core java.

Nandu says

MARCH 20, 2014 AT 12:00 PM

I Really Appreciate your work.you explained each and everything very well.

Shashikant says

MARCH 7, 2014 AT 7:26 AM

Very Helpful Question and Ans..

Madhusmita Nayak says

MARCH 2, 2014 AT 7:29 PM

What is Association, aggregation and composition in oops?

Prashanth says

MARCH 3, 2014 AT 11:14 PM

Aggregation is an association in which one class belongs to a collection. This is a part of a whole relationship where a part can exist without a whole. Aggregation has a weaker relationship.
Composition is an association in which one class belongs to a collection. This is a part of a whole relationship where a part cannot exist without a whole. If a whole is deleted then all parts are

deleted.So composition has a stronger relationship.

Madhusmita Nayak says

JUNE 6, 2014 AT 11:55 PM

Thanx for the reply !

Sindhu says

FEBRUARY 28, 2014 AT 8:39 PM

Hi Pankaj, Useful questions & simple and straight explanations □ !!

Keep going..

Regards,

Sanyhn

Ramesh Dhrath says

FEBRUARY 21, 2014 AT 12:07 PM

it's very good study material for who is beginner and also for developer…

Sanjay says

FEBRUARY 14, 2014 AT 6:26 PM

Hi Pankaj, You captured most of the interview questions, can you please cover the Java Swings , abd Java Rest API questions as well?

Letsdecode says

FEBRUARY 10, 2014 AT 5:46 PM

Thanks for Posting these questions. They are really helpful.

Just a very little correction in Ans-40.

We can use super keyword to invoke super class constructor in child class constructor but in this case it should be the FILE statement in the constructor method.

Instead of FILE it should be FIRST.

Regards

letsdecoder

Pankaj says

FEBRUARY 10, 2014 AT 7:06 PM

Thanks for replying with this small typo issue, corrected and updated the post.

Madhusmita Nayak says

FEBRUARY 9, 2014 AT 7:05 AM

What is method chaining and how it does work in java?

Manendra Singh Rawat says

JANUARY 9, 2014 AT 3:07 AM

Will you please let me know the difference between portability and platform independency because in portability we can also run the same source code on different OS.
Eg. We can run a C program on window, linux and DOS
So as per your answer "Platform independence means that you can run the same Java Program in any Operating System" platform independency and portability are same
but both are difference as per my knowledge
So will you please elaborate on this

abcd says

JANUARY 7, 2014 AT 6:12 PM

I believe in inheritance you can use access modifier to restrict subclass from using its respective methods. private won't let anyone use particular method or variable whether it is a subclass or other class in same package…

Madiraju Krishna Chaitanya says

DECEMBER 1, 2013 AT 2:58 AM

Hi Pankaj Sir,Thanks a LOT for providing this Interview material for us.

Search for tutorials...

DOWNLOAD ANDROID APP

GET IT ON Google Play

IMPORTANT INTERVIEW QUESTIONS

Java Interview Questions

- Core Java Interview Questions
- String Interview Questions
- Multithreading Interview Questions
- Collections Interview Questions
- Exception Interview Questions
- Java Programming Interview Questions
- Java 8 Interview Questions Part 1
- Java 8 Interview Questions Part 2
- Servlet Interview Questions
- JSP Interview Questions
- Struts 2 Interview Questions
- JDBC Interview Questions
- Spring Interview Questions
- Hibernate Interview Questions
- JSF Interview Questions
- Web Services Interview Questions
- Scala Basic Interview Questions
- Scala Intermediate Interview Questions
- Scala Advanced Interview Questions
- Scala Interview Questions Summary
- Common Job Interview Questions

Miscellaneous

- Java ClassLoader
- String StringBuffer StringBuilder
- Java is Pass By Value
- Java Heap vs Stack Memory

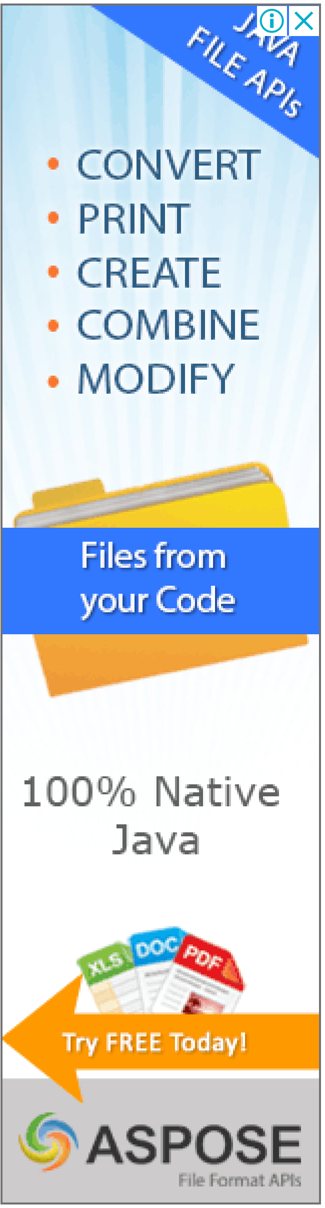
RECOMMENDED TUTORIALS

Java Tutorials

- Java IO
- Java Regular Expressions
- Multithreading in Java
- Java Logging
- Java Annotations
- Java XML
- Collections in Java
- Java Generics
- Exception Handling in Java
- Java Reflection
- Java Design Patterns
- JDBC Tutorial

Java EE Tutorials

- Servlet JSP Tutorial
- Struts2 Tutorial
- Spring Tutorial
- Hibernate Tutorial
- Primefaces Tutorial
- Apache Axis 2
- JAX-RS
- Memcached Tutorial


JAVA FILE APIs
• CONVERT
• PRINT
• CREATE
• COMBINE
• MODIFY
Files from your Code
100% Native Java
XLS
DOC
PDF
Try FREE Today!
ASPOSE
File Format APIs