

Java Interview Questions

Java Basic Interview Questions

How Java is better than C++?

Java is better than C++. Compared to C++, Java doesn't have the following jargon concepts.

- Destructor
- Goto statement
- Pointers
- Multiple Inheritance
- Operator overloading
- Template
- Scope resolution operator

How to declare constant variables in Java?

“final “ keyword is used to declare constants in Java.

The following statement declares and initializes a constant named `PI` with value `3.14` .

```
final int PI=3.14;
```

Java does not have destructors. Then how the role of destructor (free the resources that the object may have acquired) will be completed in Java?

- The role of a destructor is taken care by a predefined subroutine, that will be run by the Java runtime system.
- This is called garbage collection mechanism.
- This mechanism will be automatically invoked periodically. It will scan the memory and destroy the unused objects.
- Whenever we come out of the scope of the object, that object may not get destroyed immediately but after sometime, it will be destroyed by garbage collection mechanism.
- `System.gc()` is a method to invoke garbage mechanism explicitly.

What is a static keyword in Java? How it will be used?

The static in Java is used for memory management mainly.

The static keyword can be:

1. Static variable
2. Static method
3. Static block

Can we override static methods of a class?

We cannot override static methods. Static methods belong to a class and not to individual objects and are resolved at the time of compilation(notatruntime). Even if we try to override a static method, we will not get a compilation error nor the impact of overriding when running the code.

Is String a data type in java?

The String is not a primitive data type in java. When a string is created in java, it's actually an object of Java.Lang.String class that gets created. After creation of this string object, all built-in methods of String class can be used on the string object.

How we can execute any code even before the main method?

If we want to execute any statements even before the creation of objects at load time of class, we can use a static block of code in the class. Any statements inside this static block of code will get executed once at the time of loading the class even before the creation of objects in the main method.

What is the need for Java to be a platform independent language?

Java is a platform independent language. The need for java an independent language is that it could be used to create software to be embedded in various consumer electronic devices such as microwave ovens, remote controls etc.

How is the user-written Java code run by the computer?

Java implementations typically use a two-step compilation process. Java source code is compiled down to bytecode by the Java compiler. The bytecode is executed by a Java Virtual Machine (JVM). Modern JVMs use a technique called Just-in-Time (JIT) compilation to compile the bytecode to native instructions understood by hardware CPU on the fly at runtime.

Java program ?javac ?bytecode ?jvm ?object code

What is Just In Time compiler (JIT) ?

In the Java programming language and environment, a just-in-time(JIT) compiler is a program that turns Java bytecode into instructions that can be sent directly to the processor. After you have written a java program, the source language statements are compiled by the java compiler into bytecode rather than into code that contains instructions that match a particular hardware platform's processor. The bytecode is platform-independent code that can be sent to any platform and run on that platform.

What are Java varargs? How do they work?

Varargs in Java enables a method to accept a variable number of arguments. We use three dots(...) also known as ellipsis in the method signature to make it accept variable arguments.

- We can have only one varargs in the method definition.
- Only the last argument of a method can be varargs.
- Varargs method should not be overloaded.

When we invoke a method with variable arguments,Java compiler matches the arguments from left to right. Once it reaches to the last varargs parameter, it creates an array of the remaining arguments and passes it to the method.

What gives Java its 'write once and run anywhere' feature?

The bytecode in java give this feature to Java. Java compiler converts the Java programs into the bytecode which is the intermediate language between source code and machine code. The bytecode is not platform specific and can be executed on any computer.

If I write static public void instead of public static void, does the program compiles and runs correctly?

The program compiles and runs correctly because the order of specifier doesn't matter in Java.

What is the purpose of the default constructor in Java?

The default constructor is to assign default values to the objects. Java creates default constructor implicitly.

For integer 0 will be initialized implicitly and for string null value will be initialized.

Why constructors are not inherited in java?

-
- Constructors should have the name of the class. So if constructors were inherited in a child class, then, child class would contain a parent class constructor, which is against the constraint that constructor should have the same name as class name.
 - It is not possible to achieve encapsulation because by using super class's constructor we can access /initialize private members of the class.

What does a constructor return in java?

A constructor implicitly returns a new instance of the class it belongs to, even if it doesn't have an explicit return statement.

What can't constructors be final, static and abstract in Java?

- When you set a method as final it means: "You don't want any class override it." But the constructor can't be overridden, so it is clean.
- When you set a method as abstract it means: "The method doesn't have a body and it should be implemented in a child class." But the constructor is called implicitly when the new keyword is used so it can't lack a body.
- When you set a method as static it means: "The method belongs to the class, not a particular object." But the constructor is implicitly called to initialize an object, so there is no purpose in having a static constructor.

Why is the main method static in Java?

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

Can we execute a program without main() method?

Yes, using static block we can execute programs without main() method. static block will get executed before the main() method.

Why object class is the base class of all classes in java?

An Object class is the base class of all classes in Java. This class is in the default package of Java that is java.lang.Object. That is why you don't need to inherit this class. But each and every class in Java inherit it implicitly. This is done by the JVM.

Why Java does not support multiple inheritance?

Java does not support multiple inheritance. Suppose **C** is the child class extending from both parent class **A** and parent class **B** with some methods defined in them. Then the child class cannot understand which class

and parent class `B` with some methods defined in them. Then, the child class cannot understand which class method to call. so there is a confusion here which leads to ambiguity and leads to a compile-time error. This is the reason why Java does not support multiple inheritance.

Can a public class Myclass be defined in a source file named Yourclass.java?

No, because there is an access specifier public in class Myclass.

Do I need to import java.lang everytime?

No, because java.lang package is an inbuilt package.

What will be the output of the following expressions?

```
-14%3  
14%3
```

Answer

```
-2  
2
```

The sign of the first operand is retained in modulo operator.

What will happen if we define a concrete method in an interface?

By default interface methods are abstract. If we declare any concrete method in an interface compile-time error will come. The error will be: "Abstract methods do not specify a body".

Example

```
package sample;  
  
public interface Interview{  
    void display(){  
  
    }  
}
```

Can we create an object for an interface?

No. We cannot create an object for an interface. But, we can create a variable from an interface.

Java Packages Interview Questions

Can I import same package twice?

We can import the same package multiple times. We won't get any errors because the JVM will load the class only once no matter how many times you import the same class.

What are Java Packages? What's the significance of packages?

In Java, Package is a collection of classes and interfaces which are bundled together as they are related to each other. Use of packages helps developers to modularize the code and group the code for proper re-use. Once the code has been packaged in Packages, it can be imported into other classes and used.

Does Importing a package imports its sub-packages as well in Java?

In Java, when a package is imported, its sub-packages aren't imported and developer needs to import them separately if required. For example, if a developer imports a package `tutorial.*`, all classes in the package named `tutorial` are loaded but no classes from the sub-package are loaded. To load the classes from its sub-package (say `employ`), developer has to import it

```
import tutorial.employ.*;
```

Java Exception Handling Interview Questions

What is an exception in Java?

In Java, an exception is an abnormal condition that arises in a code sequence at runtime. An exception is a runtime error.

Whenever any error occurs while executing a java statement, an exception object is created and then JRE tries to find an exception handler to handle the exception. If a suitable exception handler is found then the exception object is passed to the handler code to process the exception, known as catching the exception. If no handler is found then application throws the exception to the runtime environment and JRE terminates the program.

Is it possible to have a standalone try block in java?

No. Standalone try block is not possible. For every try block, there should be a catch block or finally block.

```
public class Tutorial{
    public static void main(String arg[]) {
        try{

        }
    }
}
```

is not possible.

What will be the output of the following code?

```
class Example {
    public static void main(String arg[]) {
        try{

        }

        System.out.println("tutorial");

        catch(){

        }

    }
}
```

Answer

Compile Error. Because the try block and catch block should come simultaneously. There should not be any piece of code in between them.

Can we process two types of exceptions(say ArithmeticException and ArrayStoreException) with only one catch block without using generic exception(Exception)?

Yes. We can catch multiple exceptions in a single catch block using logical OR operator as shown in the following example.

```
catch(ArithmeticException | ArrayStoreException e) {
    System.out.println(e);
}
```

What will be the output when we have two finally blocks in the same class?

The program will show an error because a class should have single finally block.

What is the difference between a checked and an unchecked exception?

A checked exception must be handled within a try-catch block or declared in a throws clause; whereas an unchecked exception is not required to be handled nor declared.

Checked and unchecked exceptions are also known as compile-time and runtime exceptions respectively.

All exceptions are checked exceptions, except those indicated by Error, RuntimeException, and their subclasses.

What is exception chaining?

Occurs when an exception is thrown in response to another exception. The following example shows exception chaining:

```
import java.io.IOException;

public class ChainedException {

    public static void divide(int a, int b) {
        if(b==0) {

            ArithmeticException ae = new ArithmeticException("top layer");

            ae.initCause( new IOException("cause") );
            throw ae;
        } else {
            System.out.println(a/b);
        }
    }

    public static void main(String[] args) {
        try {
            divide(5, 0);
        } catch(ArithmeticException ae) {
            System.out.println( "caught : " +ae);
            System.out.println("actual cause: "+ae.getCause());
        }
    }
}
```

Java String Handling Interview Questions

Why Strings in Java are called as Immutable?

In java, string objects are called immutable as once value has been assigned to a string, it can't be changed and if changed, a new object is created. In below example, reference str refers to a string object having value "Value one".

```
String str="value";
```

When a new value is assigned to it, a new String object gets created and the reference is moved to the new object.

```
str="new value";
```

Is it possible to inherit the String class in java?

No. It is not possible to inherit String class in java because String is declared as final by default. So, we can't create a subclass.

Why StringBuffer and StringBuilder classes are introduced in java when there already exist String class to represent the set of characters?

The objects of String class are immutable in nature. i.e you can't modify them once they are created. If you try to modify them, a new object will be created with modified content. This may cause memory and performance issues if you are performing lots of string modifications in your code. To overcome these issues, StringBuffer and StringBuilder classes are introduced in Java.

What is the difference between equals() and "=="?

We can use == operators for reference comparison (address comparison) and equals() method for content comparison. In simple words, == checks if both objects point to the same memory location whereas equals() evaluates the comparison of values in the objects.

How to convert an integer to a string and a string to an integer in Java?

```
public class ConvertStringToInteger {
    public static void main(String[] args) {
        String str1 = "5";
        int result = Integer.parseInt(str1);
        System.out.println(result);

        String str2 = "5";
        Integer result2 = Integer.valueOf(str2);
        System.out.println(result2);
    }
}
```

What is a String pool in Java?

String pool is a special storage area in Java heap, mostly located on PerGen space, to store String literals like "ABC". When Java program creates a new String using String literal, JVM checks for that String in the pool and if String literal is already present in the pool than the same object is returned instead of creating a whole new object. String pool check is only performed when you create String as literal, if you create String using new() operator, a new String object will be created even if String with the same content is available in the pool.

Why Char array is preferred over String for storing password?

String is immutable in Java and stored in String pool. Once it's created it stays in the pool until unless garbage collected, so even though we are done with password it's available in memory for longer duration and there is no way to avoid it. It's a security risk because anyone having access to memory dump can find the password as clear text. If we use a char array to store password, we can set it to blank once we are done with it. So we can control for how long it's available in memory that avoids the security threat with String.

Different String concatenation methods in Java?

Java provides the following ways to concatenate strings :

1. Using + operator
2. Using String concat() method
3. Using StringBuffer/StringBuilder append() method
4. Using String join() method

What is the difference between String, StringBuffer, and StringBuilder classes?

Strings are immutable in Java.

StringBuffer provides a mutable alternative to String. StringBuffer operations synchronized and hence provides thread safety.

StringBuilder is another mutable alternative to String. However, it is not synchronized.

StringBuffer should be used in a multithreaded environment to avoid data corruption. StringBuilder is faster compared to StringBuffer as there is no overhead for synchronization.

Java Multithreading Interview Questions

You have threads: T1, T2, and T3. How will you ensure that thread T2 is run after T1 and thread T3 after T2?

This can be done by using `join()` method in java.

How can you say Thread behavior is unpredictable?

The solution to question is quite simple because execution of Threads depends on Thread scheduler, thread scheduler may have a different implementation on different platforms like Windows, Unix etc. Same threading program may produce a different output in subsequent executions even on the same platform.

To achieve this, we are going to create two threads on the same Runnable Object, create for loop in `run()` method and start both threads.

There is no surety that which threads will complete first, both threads will enter anonymously in for loop.

When is a thread a lightweight process or a heavyweight process in Java?

Threads are lightweight process only if threads of the same process are executing concurrently.

If threads of different processes are executing concurrently then threads are heavyweight processes.

How the threads are created in the implementation of Java?

Threads can be created using Java in two ways:

1. By creating a new class and extend this class with Thread class.
2. By implementing Runnable interface.

What is the difference between a process and a thread?

Threads are considered lightweight because they use fewer resources than processes. Threads are easier to create than processes since they do not require a separate address space.

Process is a heavyweight process and it is capable to request resources.

What is the difference between preemptive scheduling and time slicing?

Preemptive scheduling: The highest priority task executes until it enters

the waiting or dead states or a higher priority task comes into existence.

Time slicing: A task executes for a predefined slice of time and then reenters

the pool of ready tasks. The scheduler then determines which task should execute

next, based on priority and other factors.

Why methods like `wait()`, `notify()` and `notifyAll()` are present in object class and not in Thread class?

Object class has monitors which allow the thread to lock an object, while Thread does not have any monitors. When any of the above methods are called it waits for another thread to release the object and notifies the monitor by calling `notify()` or `notify all()`. When `notify()` method is called it does the job of notifying all threads which are waiting for the object to be released. The object class's monitor checks for the object if it is available or not. Thread class having these methods would not help as multiple threads exist on an object and not vice versa.

What is the difference between `sleep()` and `wait()` method in java?

Wait method releases the lock while sleep method doesn't release the lock.

Wait method belongs to `java.lang.Object` class while sleep method belongs to `java.lang.Thread` class.

What is the difference between the user thread and daemon thread?

When we create a thread in java program, then it is called as user thread. We can not make a user thread to daemon thread if a thread is started.

The daemon threads are the low priority threads that provide the background support to the user threads. It provides services to the user threads. A child thread created from a daemon thread is also a daemon thread.

What will be the output if we start a thread twice?

We can't start a thread twice. If we do so, then it will throw an Exception.

What is a deadlock?

Deadlock is a situation where two threads are waiting for each other to release locks held by them on resources. For example

Thread 1: Locks resource A, waits for resource B Thread 2: Locks resource B, waits for resource A

What is Race Condition in Java and how can we solve it?

When more than one thread tries to access the same resource without synchronization, causes a race condition. We can solve race condition by using a synchronized block or synchronized method.

What is context-switching in multi-threading?

It is the process of storing and restoring of CPU state. This helps to resume thread execution from the same point at a later point in time. It is one of the essential features for multitasking operating system and support for the multi-threaded environment.

What join() method does?

The join() method waits for a thread to die. It forces all the running threads to stop executing till the time the thread joins to complete its job.

What are the main differences between notify and notifyAll in Java?

Notify () method doesn't provide any way to choose a particular thread, that's why it's only useful when a single While notifyAll() sends a notification to all threads. It also allows them to compete for locks. It also ensures that at least one thread will proceed further.

How can we achieve thread safety in Java?

There are several ways to achieve thread safety in java – synchronization, atomic concurrent classes, implementing concurrent Lock interface, using volatile keyword, using immutable classes and Thread-safe classes.

COLLECTION FRAMEWORKS:

How does HashMap work in Java?

A HashMap in Java stores key-value pairs. The HashMap requires a hash function and uses hashCode and equals methods, in order to put and retrieve elements to and from the collection respectively. When the put method is invoked, the HashMap calculates the hash value of the key and stores the pair in the appropriate index inside the collection. If the key exists, its value is updated with the new value. Some important characteristics of a HashMap are its capacity, its load factor and the threshold resizing.

What is the difference between Array and ArrayList? When will you use Array over ArrayList?

The Array and ArrayList classes differ on the following features:

- Arrays can contain primitives or objects, while an ArrayList can contain only objects.
- Arrays have fixed size, while an ArrayList is dynamic.
- An ArrayList provides more methods and features, such as addAll, removeAll, iterator, etc.
- For a list of primitive data types, the collections use autoboxing to reduce the coding effort. However, this approach makes them slower when working on fixed size primitive data types.

What is the difference between Enumeration and Iterator interface?

Enumeration is twice as fast as Iterator and uses very less memory. Enumeration is very basic and fits basic needs. But Iterator is much safer as compared to Enumeration because it always denies other threads to modify the collection object which is being iterated by it.

Iterator takes the place of Enumeration in the Java Collections Framework. Iterators allow the caller to remove elements from the underlying collection that is not possible with Enumeration. Iterator method names have been improved to make it's functionality clear.

Why do we need collections in Java?

- When input size is dynamic.
- Collection framework is nothing but the data structure in Java. So we can use its functionality instead of writing too much code.
- Whenever you are required to store heterogeneous data.
- When data grows and shrinks frequently.

Which method is used to convert an ArrayList into Array?

toArray() method is used to convert an ArrayList into Array. toArray() method is defined in Collection interface.

What is the difference between List and a Set?

1. A list can contain duplicate values but Set doesn't allow duplicates.
2. A list allows retrieval of data to be in the same order of insertion but Set does not ensure the order in which the elements can be retrieved.

What is the difference between Comparable and Comparator interface?

Comparable and Comparator interfaces are used to sort collection or array of objects.

Comparable interface is used to provide the natural sorting of objects and we can use it to provide sorting based on single logic. Comparator interface is used to provide different algorithms for sorting and we can choose the comparator we want to use to sort the given collection of objects.

What is a fail-safe Iterator?

An iterator is considered fail-safe if it does not throw ConcurrentModificationException.

ConcurrentModificationException is not thrown as the fail-safe iterator makes a copy of the underlying structure and iteration is done over that snapshot.

Since iteration is done over a copy of the collection so interference is impossible and the iterator is guaranteed not to throw ConcurrentModificationException.

Why HashMap not allowing duplicate keys? How it is stored?

HashMap contains a hash function that takes key as parameter and returns an unique index corresponding to that key. The value related to that key is mapped with that index generated. If hashmap allows duplicate keys, then its hash function will generate the same index for two same keys and this will guarantee the case of a collision.

Although we have methods to handle collisions because it may be the case that two different keys generate the same index. But this is very rare and if that happens..we have methods such as single chaining, linear probing, quadratic probing, and double hashing to avoid those scenarios.

Due to the fact that hashmap does not allow duplicate keys, it is very useful in solving many algorithmic problems in an efficient way.

How HashMap works internally in Java?

HashMap works internally on the hashing principle. Hashing is the mechanism of assigning a unique code to a variable or attribute using an algorithm to enable easy retrieval. A true hashing mechanism should always return the same hashCode() when it is applied to the same object.

Which containers use a border layout as their default layout?

The Window, Frame and Dialog classes use a border layout as their default layout.

Java Applets, Controls, and Layout Managers Interview Questions

What is the difference between an Applet and a Java Application?

Applets are executed within a java enabled browser, but a Java application is a standalone Java program that can be executed outside of a browser. However, they both require the existence of a Java Virtual Machine (JVM). Furthermore, a Java application requires main() method but an applet does not need a main method for execution.

What is the difference between a Scrollbar and a ScrollPane?

A Scrollbar is a Component, but not a Container. A ScrollPane is a Container. A ScrollPane handles its own events and performs its own scrolling.

Why should we have Layout managers in java?

We have Layout managers in Java because of two reasons. They are:

- Manual positioning of control is a tedious process.
- Every time we can't provide height and width as data into the program.

What are the limitations of a Grid Layout?

In Grid Layout, we can't add controls of different sizes into the frame. To overcome this limitation, we use GridBagLayout.

How can the controls be manually positioned on the frame?

Using setBounds() method, we can manually position the controls on the Frame.

What is the difference between TextField and TextArea?

- Textfield won't have a scrollbar attached with it whereas TextArea can have a scrollbar attached with it.
- TextField can accept only a single line of input whereas TextArea can accept any lines of input.

Example Programs

1. String Palindrome
2. Avoiding deadlock in Java
3. Factorial
4. Fibonacci
5. Reversing a String

6. String Anagram
7. Reversing a Linked List
8. Implementing Queue using Stack
9. All sorting programs (merge sort, insertion sort, quick sort, bucket sort, radix sort, counting sort, heap sort, bubble sort)
10. Find the largest substring without repeating characters.
11. Producer-consumer problem
12. Priority Queue
13. TCP and UDP programs

Java Tutorial

- ◆ [Java Tutorial](#)
- ◆ [Java Introduction](#)
- ◆ [Java Installation](#)
- ◆ [IDEs for Java Development](#)
- ◆ [Java HelloWorld Program](#)
- ◆ [Java Program Structure](#)
- ◆ [Java Datatypes](#)
- ◆ [Java Variable Types](#)
- ◆ [Java Access Modifiers](#)
- ◆ [Java Operators](#)
- ◆ [Java Decision Making](#)
- ◆ [Java Loops](#)
- ◆ [Java Array](#)
- ◆ [Java OOPs](#)
- ◆ [Java String](#)
- ◆ [Java Exception Handling](#)
- ◆ [Java File Operations](#)
- ◆ [Java Date & Time](#)
- ◆ [Java MySQL](#)
- ◆ [Java Random](#)
- ◆ [Java Math](#)