Knowledge Base   Resources   Deals   Job Board   Join Us   About

Search...

ANDROID   JAVA   JVM LANGUAGES   SOFTWARE DEVELOPMENT   AGILE   CAREER   COMMUNICATIONS   DEVOPS   META JCG

🏠 Home » Java » Core Java » Core Java Interview Questions

ABOUT SUNIL GULABANI

Sunil Gulabani is a JAVA software engineer and Cloud Technology Savvy. He has developed enterprise solutions using Java(EE), Apache Solr, GWT, GWT-P, SmartGWT, Restlet and Hibernate. He holds a keen interest in system architecture and integration, data modeling, relational databases and mapping with NoSQL for high throughput.

Core Java Interview Questions

👤 Posted by: Sunil Gulabani   📁 in Core Java   🕐 September 3rd, 2013

Tip
Are you looking for career opportunities in Java?

Visit our **Job Board** to search and review a **hand-picked** selection of Java jobs for IT professionals around the world.

Additionally you can add your resume to our **Resumes Board** promoting your skills to a wide range of Java job recruiters around the world.

Q. What if the main method is declared as private?

Answer:

The program compiles properly but at runtime it will give "Main method not public." message.

Q. What is meant by pass by reference and pass by value in Java?

Answer:

Pass by reference means, passing the **address** itself rather than passing the value. Pass by value means passing a **copy** of the value.

Q. If you're overriding the method equals() of an object, which other method you might also consider?

Answer:

hashCode()

" Q. What is Byte Code?

Or

Q. What gives java it's "write once and run anywhere"

nature?

Answer:

All Java programs are compiled into class files that contain bytecodes. These byte codes can be run in any platform and hence java is said to be platform independent.

Q. Expain the reason for each keyword of public static void main(String args[])?

Answer:

- **public** – main(..) is the first method called by java environment when a program is executed so it has to accessible from java environment. Hence the access specifier has to be public.
- **static** : Java environment should be able to call this method without creating an instance of the class , so this method must be declared as static.
- **void** : main does not return anything so the return type must be void

The argument String indicates the argument type which is given at the command line and arg is an array for string given during command line.

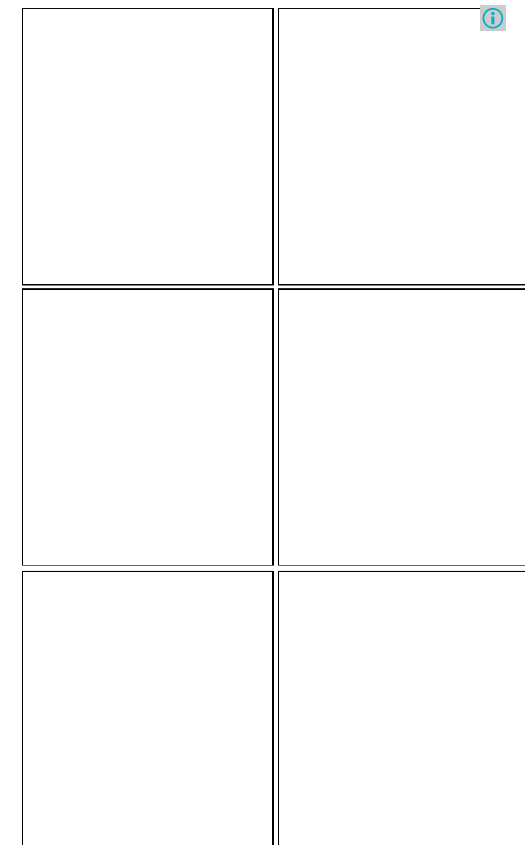> *Q. What are the differences between == and .equals() ?*

Or

Q. what is difference between == and equals

Or

Q. Difference between == and equals method

Q. What would you use to compare two String

variables – the operator == or the method equals()?

Or

Q. How is it possible for two String objects with identical values not to be equal under the == operator?

Answer:

The == operator compares two objects to determine if they are the same object in memory i.e. present in the same memory location. It is possible for two String objects to have the same value, but located in different areas of memory.

== compares references while .equals compares contents. The method public boolean equals(Object obj) is provided by the Object class and can be overridden. The default implementation returns true only if the object is compared with itself, which is equivalent to the equality operator == being used to compare aliases to the object. String, BitSet, Date, and File override the equals() method. For two String objects, value equality means that they contain the same character sequence. For the Wrapper classes, value equality means that the primitive values are equal.

```
01  public class EqualsTest {
02
03          public static void main(String[] args) {
04
05                      String s1 = "abc";
06                      String s2 = s1;
07                      String s5 = "abc";
08                      String s3 = new String("abc");
09                      String s4 = new String("abc");
10                      System.out.println("== comparison : " + (s1 == s5));
11                      System.out.println("== comparison : " + (s1 == s2));
12                      System.out.println("Using equals method : " + s1.equals(s2));
13                      System.out.println("== comparison : " + s3 == s4);
14                      System.out.println("Using equals method : " + s3.equals(s4));
15          }
16  }
```

Output

```
1  == comparison : true
2  == comparison : true
3  Using equals method : true
4  false
5  Using equals method : true
```

Q. What if the static modifier is removed from the

> ## signature of the main method?
>
> Or

Q. What if I do not provide the String array as the argument to the method?

Answer:

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

Q. Why oracle Type 4 driver is named as oracle thin driver?

Answer:

Oracle provides a Type 4 JDBC driver, referred to as the Oracle "thin" driver. This driver includes its own implementation of a TCP/IP version of Oracle's Net8 written entirely in Java, so it is platform independent, can be downloaded to a browser at runtime, and does not require any Oracle software on the client side. This driver requires a TCP/IP listener on the server side, and the client connection string uses the TCP/IP port address, not the TNSNAMES entry for the database name.

> ## Q. What is the difference between final, finally and finalize? What do you understand by the java final keyword?
>
> Or

Q. What is final, finalize() and finally?

Or

Q. What is finalize() method?

Or

Q. What does it mean that a class or member is final?

Answer:

- **final** – declare constant
- **finally** – handles exception
- **finalize** – helps in garbage collection

Variables defined in an interface are implicitly final. A final class can't be extended i.e., final class may not be subclassed. This is done for security reasons with basic classes like String and Integer. It also allows the compiler to make some optimizations, and makes thread safety a little easier to achieve. A final method can't be overridden when its class is inherited. You can't change value of a final variable (is a constant). finalize() method is used just before an object is destroyed and garbage collected. finally, a key word used in exception handling and will be executed whether or not an exception is thrown. For example, closing of open connections is done in the finally method.

Q. What is the Java API?

Answer:

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets.

Q. What is the GregorianCalendar class?

Answer:

The GregorianCalendar provides support for traditional Western calendars.

Q. What is the ResourceBundle class?

Answer:

The ResourceBundle class is used to store locale-specific resources that can be loaded by a program to tailor the program's appearance to the particular locale in which it is being run.

Q. Why there are no global variables in Java?

Answer:

Global variables are globally accessible. Java does not support globally accessible variables due to following reasons:

- The global variables breaks the <u>referential transparency</u>
- Global variables create <u>collisions in namespace</u>.

Q. How to convert String to Number in java program?

Answer:

The valueOf() function of Integer class is is used to convert string to Number. Here is the code example:

```
1  String numString = "1000";
2  int id=Integer.valueOf(numString).intValue();
```

Q. What is the SimpleTimeZone class?

Answer:

The SimpleTimeZone class provides support for a Gregorian calendar.

Q. What is the difference between a while statement and a do statement?

Answer:

A while statement (pre test) checks at the beginning of a loop to see whether the next loop iteration should occur. A do while statement (post test) checks at the end of a loop to see whether the next iteration of a loop should occur. The do statement will always execute the loop body at least once.

Q. What is the Locale class?

Answer:

The Locale class is used to tailor a program output to the conventions of a particular geographic, political, or cultural region.

Q. Describe the principles of OOPS.

Answer:

There are three main principals of oops which are called Polymorphism, Inheritance and Encapsulation.

Q. Explain the Inheritance principle.

Answer:

Inheritance is the process by which one object acquires the properties of another object. Inheritance allows well-tested procedures to be reused and enables changes to make once and have effect in all relevant places

Q. What is implicit casting?

Answer:

Implicit casting is the process of simply assigning one entity to another without any transformation guidance to the compiler. This type of casting is not permitted in all kinds of transformations and may not work for all scenarios.

Example

```
1  int i = 1000;
2  long j = i; //Implicit casting
```

Q. Is sizeof a keyword in java?

Answer:

The sizeof is not a keyword.

Q. What is a native method?

Answer:

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

Q. In System.out.println(), what is System, out and println?

Answer:

System is a predefined final class, out is a PrintStream object and println is a built-in overloaded method in the out object.

> ## Q. What are Encapsulation, Inheritance and Polymorphism
>
> Or
>
> ## Q. Explain the Polymorphism principle. Explain the different forms of Polymorphism.

Answer:

Polymorphism in simple terms means <u>one name many forms</u>. Polymorphism enables one entity to be used as a general category for different types of actions. The specific action is determined by the exact nature of the situation.

Polymorphism exists in three distinct forms in Java:

- Method overloading
- Method overriding through inheritance
- Method overriding through the Java interface

Q. What is explicit casting?

Answer:

Explicit casting in the process in which the complier are specifically informed to about transforming the object.

Example

```
1  long i = 700.20;
2  int j = (int) i; //Explicit casting
```

Q. What is the Java Virtual Machine (JVM)?

Answer:

The Java Virtual Machine is software that can be ported onto various hardware-based platforms

Q. What do you understand by downcasting?

Answer:

The process of Downcasting refers to the casting from a general to a more specific type, i.e. casting down the hierarchy

> ## Q. What are Java Access Specifiers?

Or

Q. What is the difference between public, private, protected and default Access Specifiers?

Or

Q. What are different types of access modifiers?

Answer:

Access specifiers are keywords that determine the type of access to the member of a class. These keywords are for allowing privileges to parts of a program such as functions and variables. These are:

- Public: accessible to all classes
- Protected: accessible to the classes within the same package and any subclasses.
- Private: accessible only to the class to which they belong
- Default: accessible to the class to which they belong and to subclasses within the same package

Q. Which class is the superclass of every class?

Answer:

Object.

Q. Name primitive Java types.

Answer:

The 8 primitive types are byte, char, short, int, long, float, double, and boolean. Additional is String.

" " Q. What is the difference between static and non-static variables?

Or

Q. What are "class variables"?

Or

Q. What is static in java?

Or

Q. What is a static method?

Answer:

A static variable is associated with the class as a whole rather than with specific instances of a class. Each object will share a common copy of the static variables i.e. there is only one copy per class, no matter how many objects are created from it. Class variables or static variables are declared with the static keyword in a class. These are declared outside a class and stored in static memory. Class variables are mostly used for constants. Static variables are always called by the class name. This variable is created when the program starts and gets destroyed when the programs stops. The scope of the class variable is same an instance variable. Its initial value is same as instance variable and gets a default value when it's not initialized corresponding to the data type. Similarly, a static method is a method that belongs to the class rather than any object of the class and doesn't apply to an object or even require that any objects of the class have been instantiated. Static methods are implicitly final, because overriding is done based on the type of the object, and static methods are attached to a class, not an object. A static method in a superclass can be shadowed by another static method in a subclass, as long as the original method was not declared final. However, you can't override a static method with a non-static method. In other words, you can't change a static method into an instance method in a subclass.

Non-static variables take on unique values with each object instance.

Q. What is the difference between the boolean & operator and the && operator?

Answer:

If an expression involving the boolean & operator is evaluated, both operands are evaluated, whereas the && operator is a short cut operator. When an expression involving the && operator is evaluated, the first operand is evaluated. If the first operand returns a value of true then the second operand is evaluated. If the first operand evaluates to false, the evaluation of the second operand is skipped.

Q. How does Java handle integer overflows and underflows?

Answer:

It uses those low order bytes of the result that can fit into the size of the type allowed by the operation.

Q. What if I write static public void instead of public static void?

Answer:

Program compiles and runs properly.

Q. What is the difference between declaring a variable and defining a variable?

Answer:

In declaration we only mention the type of the variable and its name without initializing it. Defining means declaration + initialization. E.g. String s; is just a declaration while String s = new String ("bob"); Or String s = "bob"; are both definitions.

Q. What type of parameter passing does Java support?

Answer:

In Java the arguments (primitives and objects) are always **passed by value.** With objects, the object reference itself is passed by value and so both the original reference and parameter copy both refer to the same object.

Q. Explain the Encapsulation principle.

Answer:

Encapsulation is a process of binding or wrapping the data and the codes that operates on the data into a single entity. This keeps the data safe from outside interface and misuse. Objects allow procedures to be encapsulated with their data to reduce potential interference. One way to think about encapsulation is as a protective wrapper that prevents code and data from being arbitrarily accessed by other code defined outside the wrapper.

Q. What do you understand by a variable?

Answer:

Variable is a named memory location that can be easily referred in the program. The variable is used to hold the data and it can be changed during the course of the execution of the program.

Q. What do you understand by numeric promotion?

Answer:

The Numeric promotion is the conversion of a smaller numeric type to a larger numeric type, so that integral and floating-point operations may take place. In the numerical promotion process the byte, char, and short values are converted to int values. The int values are also converted to long values, if necessary. The long and float values are converted to double values, as required.

Q. What do you understand by casting in java language? What are the types of casting?

Answer:

The process of converting one data type to another is called Casting. There are two types of casting in Java; these are implicit casting and explicit casting.

Q. What is the first argument of the String array in main method?

Answer:

The String array is empty. It does not have any element. This is unlike C/C++ where the first element by default is the program name. If we do not provide any arguments on the command line, then the String array of main method will be empty but not null.

Q. How can one prove that the array is not null but empty?

Answer:

Print array.length. It will print 0. That means it is empty. But if it would have been null then it would have thrown a NullPointerException on attempting to print array.length.

Q. Can an application have multiple classes having main method?

Answer:

Yes. While starting the application we mention the class name to be run. The JVM will look for the main method only in the class whose name you have mentioned. Hence there is not conflict amongst the multiple classes having main method.

Q. When is static variable loaded? Is it at compile time or runtime? When exactly a static block is loaded in Java?

Answer:

Static variable are loaded when classloader brings the class to the JVM. It is not necessary that an object has to be created. Static variables will be allocated memory space when they have been loaded. The code in a static block is loaded/executed only once i.e. when the class is first initialized. A class can have any number of static blocks. Static block is not member of a class, they do not have a return statement and they cannot be called directly. Cannot contain this or super. They are primarily used to initialize static fields.

Q. Can I have multiple main methods in the same class?

Answer:

We can have multiple overloaded main methods but there can be **only one main method with the following signature :**

```
1  public static void main(String[] args) {}
```

No the program fails to compile. The compiler says that the main method is already defined in the class.

Q. Explain working of Java Virtual Machine (JVM)?

Answer:

JVM is an abstract computing machine like any other real computing machine which first converts .java file into .class file by using Compiler (.class is nothing but byte code file.) and Interpreter reads byte codes.

Q. How can I swap two variables without using a third variable?

Answer:

Add two variables and assign the value into First variable. Subtract the Second value with the result Value. and assign to Second variable. Subtract the Result of First Variable With Result of Second Variable and Assign to First Variable. Example:

```
1  int a=5,b=10;a=a+b; b=a-b; a=a-b;
```

An other approach to the same question

You use an XOR swap. (BEST APPROACH) as in case of using above approach it may goes over/under flow. For example:

```
1  int a = 5; int b = 10;
2  a = a ^ b;
3  b = a ^ b;
4  a = a ^ b;
```

Q. What is data encapsulation?

Answer:

Encapsulation may be used by creating 'get' and 'set' methods in a class (JAVABEAN) which are used to access the fields of the object. Typically the fields are made private while the get and set methods are public. Encapsulation can be used to validate the data that is to be stored, to do calculations on data that is stored in a field or fields, or for use in introspection (often the case when using javabeans in Struts, for instance). Wrapping of data and function into a single unit is called as data encapsulation. Encapsulation is nothing but wrapping up the data and associated methods into a single unit in such a way that data can be accessed with the help of associated methods. Encapsulation provides data security. It is nothing but data hiding.

Q. What is reflection API? How are they implemented?

Answer:

Reflection is the process of introspecting the features and state of a class at runtime and dynamically manipulate at run time. This is supported using Reflection API with built-in classes like Class, Method, Fields, Constructors etc. Example: Using Java Reflection API we can get the class

name, by using the getName method.

Q. Does JVM maintain a cache by itself? Does the JVM allocate objects in heap? Is this the OS heap or the heap maintained by the JVM? Why

Answer:

Yes, the JVM maintains a cache by itself. It creates the Objects on the HEAP, but references to those objects are on the STACK.

Q. What is phantom memory?

Answer:

Phantom memory is false memory. Memory that does not exist in reality.

Q. Can a method be static and synchronized?

Answer:

A static method can be synchronized. If you do so, the JVM will obtain a lock on the java.lang.

Class instance associated with the object. It is similar to saying:

```
1  synchronized(XYZ.class) {
2  }
```

Q. What is difference between String and StringTokenizer?

Answer:

A StringTokenizer is utility class used to break up string.

Example:

```
1  StringTokenizer st = new StringTokenizer("Hello World");
2  while (st.hasMoreTokens()) {
3  System.out.println(st.nextToken());
4  }
```

Output:

```
1  Hello
2  World
```

Question: What is transient variable?

Answer:

Transient variable can't be serialize. For example if a variable is declared as transient in a Serializable class and the class is written to an ObjectStream, the value of the variable can't be written to the stream instead when the class is retrieved from the ObjectStream the value of the variable becomes **null**.

Note

```
01  transient
02
03  identifies a variable not to be written out when an
04
05  instance is serialized  (It can't be copied to remove
06
07  system)
08
09
10
11  volatile
12
13  indicates that the field is used by synchronized threads
14
15  and that the compiler should not attempt to perform
16
17  optimizations with it.
18
19
20
21  When more than one thread share a (volatile) data it is
22
23  checked every time. Every thread keeps the latest value of volatile variable
```

Question: Name the containers which uses Border Layout as their default layout?

Answer:

Containers which uses Border Layout as their default are: window, Frame and Dialog classes.

Question: What do you understand by Synchronization?

Answer:

Synchronization is a process of controlling the access of shared resources by the multiple threads in such a manner that only one thread can access one resource at a time. In non synchronized multithreaded application, it is possible for one thread to modify a shared object while another thread is in the process of using or updating the object's value. Synchronization prevents such type of data corruption.

E.g. Synchronizing a function:

```
1  public synchronized void Method1 () {
2
3  // Appropriate method-related code.
4
5  }
```

E.g. Synchronizing a block of code inside a function:

```
1  public myFunction (){
2      synchronized (this) {
3            // Synchronized code here.
4          }
5  }
```

Reference: Core Java Interview Questions from our JCG partner Sunil Gulabani at the Sunil Gulabani blog.

Tagged with: INTERVIEW

Do you want to know how to develop your skillset to become a Java Rockstar?

Subscribe to our newsletter to start Rocking right now!

To get you started we give you our best selling eBooks for FREE!

1. JPA Mini Book
2. JVM Troubleshooting Guide
3. JUnit Tutorial for Unit Testing
4. Java Annotations Tutorial
5. Java Interview Questions
6. Spring Interview Questions
7. Android UI Design