**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. 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 implicit & explicit 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**

int i = 1000;

long j = i; //Implicit casting

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

**Example**

[view source](http://www.javacodegeeks.com/2013/09/core-java-interview-questions.html#viewSource)

[print](http://www.javacodegeeks.com/2013/09/core-java-interview-questions.html#printSource)[?](http://www.javacodegeeks.com/2013/09/core-java-interview-questions.html#about)

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

## Q. What is a native method?

#### Answer:

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

## 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 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. 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. 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:

[view source](http://www.javacodegeeks.com/2013/09/core-java-interview-questions.html#viewSource)

[print](http://www.javacodegeeks.com/2013/09/core-java-interview-questions.html#printSource)[?](http://www.javacodegeeks.com/2013/09/core-java-interview-questions.html#about)

|  |  |  |
| --- | --- | --- |
| 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:**

[view source](http://www.javacodegeeks.com/2013/09/core-java-interview-questions.html#viewSource)

[print](http://www.javacodegeeks.com/2013/09/core-java-interview-questions.html#printSource)[?](http://www.javacodegeeks.com/2013/09/core-java-interview-questions.html#about)

|  |  |  |
| --- | --- | --- |
| 1 | StringTokenizer st = new StringTokenizer(”Hello World”); | |
| 2 | while (st.hasMoreTokens()) { |

|  |  |  |
| --- | --- | --- |
| 3 | System.out.println(st.nextToken()); | |
| 4 | } |

**Output:**

[view source](http://www.javacodegeeks.com/2013/09/core-java-interview-questions.html#viewSource)

[print](http://www.javacodegeeks.com/2013/09/core-java-interview-questions.html#printSource)[?](http://www.javacodegeeks.com/2013/09/core-java-interview-questions.html#about)

|  |  |
| --- | --- |
| 1 | Hello |
| 2 | World |