# General

1. Java Integer Cache and why we should use **equals** instead of **==**

Integer objects are cached internally and reused via the same referenced objects.

This is applicable for Integer values in the range between –128 to +127.

This Integer caching works only on auto-boxing. Integer objects will not be cached when they are built using the constructor.

<https://javaconceptoftheday.com/why-128-128-returns-false-in-java/>

<https://www.geeksforgeeks.org/java-integer-cache/>

1. Integer.MIN\_VALUE vs Double.MIN\_VALUE

Integer.MIN\_VALUE gives the smallest Integer including negatif number, while Double.MIN\_VALUE store the smallest positive nonzero value of type double ‘2^(-1074)’.

Read more: <https://www.java67.com/2012/09/top-10-tricky-java-interview-questions-answers.html#ixzz7H8cOTwV8>

1. Private and static method overriding

We cannot override a static method because static method are binded at compile time while overriding is done dynamically at runtime.

The same goes to private method, only this time, the overriding is impossible since the private method is only visible fromwithin the class.

Read more: <https://www.java67.com/2012/09/top-10-tricky-java-interview-questions-answers.html#ixzz7H8cOTwV8>  
  
We still can create a method with the same signature in the child class, this is called method hiding.

1. What is the difference between StringBuffer and StringBuilder in Java

StringBuffer methods like length(), capacity(), or append() are [synchronized](http://javarevisited.blogspot.sg/2011/04/synchronization-in-java-synchronized.html) while corresponding methods in StringBuilder are not synchronized.  
  
Read more: <https://www.java67.com/2012/09/top-10-tricky-java-interview-questions-answers.html#ixzz7H8cOTwV8>

1. Difference between error and exception

Errors in a program are irrecoverable. Exemple (OutOfMemoryError, StackOverflowError)

In the other hand, exception can be handeled.

1. String Literal vs String Object

When we create a String object using the new() operator, it always creates a new object in heap memory. On the other hand, if we create an object using String literal syntax e.g. “Baeldung”, it may return an existing object from the String pool, if it already exists. Otherwise, it will create a new String object and put in the string pool for future re-use.

1. try-with-resources

To use try-with-resources the resource must implement AutoClosable Interface.

When using it, i twill garantie that the used resource is closed after the execution oft he block.

1. Marker Interface

Marker interface is an interface which is empty.   
Todo : Add more informations

1. Java constructors and initialization blocks

constructors and initialization blocks. Instance Initialization Blocks or IIB are used to initialize instance variables . So firstly, constructor is invoked and the java compiler copies the instance initializer block in the constructor after the first statement super(). They run each time when object of the class is created.

* Initialization blocks are executed whenever the class is initialized and before constructors are invoked.
* They are typically placed above the constructors within braces.
* It is not at all necessary to include them in your classes.

Order of execution:

* Instance Initialization Block of the superclass.
* Constructors of the superclass.
* Instance Initialization Blocks of the class.
* Constructors of the class.

<https://www.geeksforgeeks.org/instance-initialization-block-iib-java/>

1. Why Java doesn't support multiple inheritance

Avoid the ambiguity around the Diamond problem.   
Multiple inheritances does complicate the design and creates problem during casting, constructor chaining etc

Read more: <https://javarevisited.blogspot.com/2011/07/why-multiple-inheritances-are-not.html#ixzz7HnFdMsBl>

1. Optionals

To use in return statement. Don’t use an optional just to use the method if exist. Sometimes an « != null » is better.

Optional use four time more space.

1. Synchronized vs volatile vs atomic

Todo