

## Java quiz questions!

1. A method in a class declared as static can only access static class members.  
**True**, Static methods may, however, receive objects as parameters.
2. A method that is overridden in the subclass must retain the same return type and parameter list.  
**True**, The overriding method may, however, have a different throws clause as long as it only reduces the number of exceptions being thrown in the overridden method.
3. The access level of an overridden method cannot be changed in a subclass.  
**False**, The overriding method can increase the access to the overridden method. For instance, a protected method may be made public, but not private.
4. Java does not allow a method with the same signature in a subclass, as a method in the super class.  
**False**, When a method in a subclass had the same signature as a method in a parent class, then it is said to “override” that method.
5. When a method or a variable in a class is declared as private, it can only be accessed by the methods within the same class.  
**True**, There are two more access types: members declared as public can be accessed even outside the class, whereas members with default access declarataion can only be accessed within the same package.
6. A method in a class declared as static may be invoked simply by using the name of the method alone.  
**False**, This is true when the call is made from within the same class. However, when a static method is invoked from another class, then both the classname as well as the method name must be used to make the call.
7. The return value from a method must always match the declared return type.  
**True**, When a method returns an object, it must be an instance of either the exact class declared as the return type, or its subclass.
8. Variables declared inside a for loop are limited in scope to the loop.  
**False**, Any variable declared within a block statement such as a for or if cannot be referenced outside the block.
9. The expression (y >= z && a == b) is evaluated by first evaluating the expression y >= z, and then evaluating a == b.  
**False**, If y >= z is false, then there is no need to evaluate the second expression.

10. All bitwise operations are carried out with the same level of precedence in Java.  
**False**, All operations in Java, including the bitwise operations, are carried out with a definite precedence.
11. consider the statement “ $x = (a > b) ? a : b$ ”; then the value of  $x$  is 27, if  $a = 18$  and  $b = 27$ .  
**True**, the statement is equivalent to: if  $(a > b)$   $x = a$ ; else  $x = b$ ;
12. A break statement must always be present in the default case of a “switch” selection structure.  
**False**, The break statement, which brings the computation out of the “switch” selection structure, is not required for the default case.
13. The operations  $y \gg 3$  and  $y \ggg 3$  produce the same result when  $y > 0$ .  
**True**, The shift operation “ $y1 \ggg y2$ ” is identical to “ $y1 \gg y2$ ” for all positive values of  $y1$ . It shifts the bits in  $y1$  to the right by  $y2$  positions.
14. The modulus operator (%) in Java can be used only with variables of integer type.  
**False**, The modulus operator (%) may be used with floating-point as well as integer types. It returns the remainder of a division operation, e.g.,  $10 \% 6$  will return 4.
15. Objects of a subclass can be assigned to a super class reference.  
**True**, Objects of a super class may not be assigned to a sub class reference.  
 Food for thought: why is it so?
16. Declarations must appear at the start of the body of a Java method.  
**False**, They can appear anywhere within the body of the method.
17. The == operator can be used to compare two String objects. The result is always true if the two strings are identical.  
**False**, String objects must be compared using the “equals” method of one of the objects. Food for thought: will the == operator ever return true when two string objects are compared using it?
18. A break statement must always be present in the default case of a “switch” selection structure.  
**False**, The break statement, which brings the computation out of the “switch” selection structure, is not required for the default case.
19. An array in the Java programming language has the ability to store many different types of values.  
**False**, All elements of an array must be of the same type. However, it is possible to declare an array of objects, which may be instances of different classes.

20. Whenever the “&&” operator is used, such as in: `exp1 && exp2` where `exp1` and `exp2` are boolean expressions, both the boolean expressions are not always evaluated.  
**True**, If the first expression is false, the result of the “&&” operation will always be false regardless of the second expression. The “&” operator on the other hand forces the evaluation of the second expression even if the first expression is false.
21. The operations `y >> 3` and `y >>> 3` produce the same result when `y > 0`.  
**True**, The shift operation “`y1 >>> y2`” is identical to “`y1 >> y2`” for all positive values of `y1`. It shifts the bits in `y1` to the right by `y2` positions.