FINAL IN JAVA:<http://javarevisited.blogspot.com/2016/09/21-java-final-modifier-keyword-interview-questions-answers.html#axzz4pYJxoTvh>

**1) What is the use of the final keyword in Java?**

The final keyword can be used with a class, method, and variables. If it is used with class then it prevents inheritance by not allowing you to create subclasses. If it is used with methods then it prevents overriding, you cannot override a final method in Java. If it is used with variables then they are treated as constant because you cannot change their value once assigned.

**2) Can we make a variable final in Java? What is different between a normal variable and final variable?**

Yes, you can make a variable final in Java. The difference between normal variable and final variable comes from multiple assignments. you can re-assign value to a normal variable but you cannot change the value of a final variable once assigned.

**5) What is a blank final field or variable in Java?**

Java allows you to create a final member variable without assignment but requires you to assign the value in either static initializer block, if member variable is static or inside every constructor, if member variable is non-static. The final variable without assignment or value is called blank final variable, the compiler will ensure that this field is initialized in every path of execution.

**6) Can you change the state of the object to which a final reference variable is pointing?**

Yes, you can change the state of the object referred by a final variable. This is one of the tricky concept in Java and often cause subtle errors. One of the most common examples of this is Collection classes e.g. ArrayList or HashMap referenced by a final variable. You can still add, remove and update elements but you cannot change the final variable to point to another collection.

**7) Can we make an array final in Java? Can you change its elements?**

Yes, you can make an array final in Java and you can change it's elements as well. This is actually the follow-up to the previous question, both array and collection classes can be made final and you can still change their elements.

**8) Can you make a Collection final in Java e.g. ArrayList? What is the impact?**

Yes, you can make a Collection final in Java. The impact is nothing but the final variable cannot be swapped with another Collection, but you can still add, remove and update elements in ArrayList or any collection classes.

**9) What is the difference between abstract method and final method in Java?**

The abstract method is incomplete while the final method is regarded as complete. The only way to use an abstract method is by overriding it, but you cannot override a final method in Java. You should also remember that a method cannot be both abstract and final in Java because both are opposite to each other.

**10) What is the use of final class in Java?**

You make a class final when you think it's complete and nobody should alter the feature by creating a subclass. Generally, security sensitive classes are made final in Java e.g. String. Another reason is performance, compiler, and [JIT](http://www.java67.com/2013/02/difference-between-jit-and-jvm-in-java.html) both can make a lot of assumption if a class is final because they know overriding or polymorphism will not come into the picture.

**11) Can you overload a final method in Java?**

Yes, you can overload a final method in Java, remember overloading is different than overriding and you only need to declare the method with the same name in the same class but different method signature for overloading

**12) Can you override a final method in Java?**

No, you cannot override a final in Java. You make a method final in Java to prevent overriding, so no question of further allowing overriding.

**13) Can we make a static method final in Java?**

Yes, you can make a static method final in Java, nothing prevents you making a [static method](http://www.java67.com/2016/04/difference-between-static-vs-non-static-method-in-java.html) from being a final one as well. In fact, they both go hand in hand.

**16) Can you declare Constructor as final in Java?**

No, Constructors cannot be made final in Java. The compiler will throw an error if you try to make a constructor final in Java.

**17) What is constant in Java?**

A static final variable is known as constant in Java. They are also known as a compile time constant because of their value at the time of compilation. They are also inlined at the client end, means if you are using a static final variable then its value will be copied to your class at compile time. Which also means that you need to recompile all the classes which use the static final variable, whenever you change the value of a static final field

**18) Are static final variables are thread-safe?**

There are two types of static final variables, primitive, and reference. Initialization of all static final variables is thread safe because it's done in static initializer block. Similarly primitive static final variable is also thread-safe because you cannot modify their value once created, but reference static final variable may or may not be thread-safe. If the object to which your final variable is referring is [Immutable](http://javarevisited.blogspot.com/2013/03/how-to-create-immutable-class-object-java-example-tutorial.html#uds-search-results) or [thread-safe](http://www.java67.com/2015/09/thread-safe-singleton-in-java-using-double-checked-locking-pattern.html) then it is otherwise not.

**19) Can a class be abstract and final at the same time?**

No, it's not possible because the only way to use abstract class is by extending it and creating a concrete subclass, while it's not possible to extend a final class in Java.

**20) When to make a method final in Java?**

You make a method final when you know that it's complete and you want to ensure that it should not be overridden. One of the examples of a final method is [template methods](http://javarevisited.blogspot.com/2013/12/when-to-make-method-final-in-java.html) from Template design pattern, which outlines the algorithm. That method should be final so that sub-classes cannot change the algorithm, the

**21) When to make a class final in Java?**

You make a class final when you don't want anyone should extend it. This is mainly done due to security reason because it also hampers extensibility of your program. A couple of examples of final classes in JDK is String, Integer, and other wrapper class. See [here](http://www.java67.com/2014/01/why-string-class-has-made-immutable-or-final-java.html) to learn more about why String class is made final in Java.