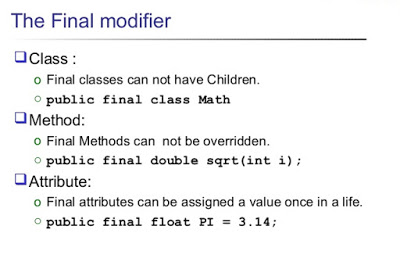
**1) What is the use of the final keyword in Java?**  
The final keyword can be used with a class, method, and variables. If it is used with class then it prevents inheritance by not allowing you to create subclasses. If it is used with methods then it prevents overriding, you cannot override a final method in Java. If it is used with variables then they are treated as constant because you cannot change their value once assigned. See [here](http://javarevisited.blogspot.com/2011/12/final-variable-method-class-java.html) to learn more about final modifier in Java.

**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. See [here](http://www.java67.com/2015/07/how-to-use-final-keyword-in-java-example.html) to learn more about when to make a variable final in Java.  
  
  
**3) Can we make a method final in Java? explain the difference between normal method and final method**  
Yes, you can make a method final in Java. The difference is in the fact that you can override a non-final method, provided it'st not private and static but you cannot override a final method in Java. See [here](http://javarevisited.blogspot.com/2013/12/when-to-make-method-final-in-java.html) to learn more about when to make a method final in Java.  
  
  
**4) Can we make local variable final in Java?**  
Yes, you can make local variable final in Java. In fact, this was mandatory, if you want to access the local variable inside an Anonymous class until Java 8. From Java 8 onward, you don't need to make it final but make sure you don't change the value once assigned. This is also known as an effectively final variable in Java. See [Java SE 8 for Really Impatient](http://www.amazon.com/Java-SE8-Really-Impatient-Course/dp/0321927761?tag=javamysqlanta-20) by Cay S. Horstmann to learn more about the effectively final variable in Java.

**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. See [here](http://www.java67.com/2012/09/what-is-blank-final-variable-in-java.html) to learn more about the blank final variable in Java.  
  
  
**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. This is different then [immutable or read-only collection](http://www.java67.com/2012/08/how-to-create-read-only-collection-in-java-example.html) in Java, where you cannot perform any add, remove or update operation once created.  
  
  
**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. This is again related to previous two questions. See [Big Java: Early Objects](https://www.amazon.com/Big-Java-Cay-S-Horstmann/dp/111842297X?tag=javamysqlanta-20) to learn more about this concept.  
  
  
**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.

[](https://2.bp.blogspot.com/-VGdW3xuDGDc/V-XuZevB-VI/AAAAAAAAHCE/ja04xO2JgQEJgfEuDa7otIDcXETBRYmGgCEw/s1600/final+modifier+in+Java.jpg)

**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. See [rules of method overloading](http://www.java67.com/2012/09/what-is-rules-of-overloading-and-overriding-in-java.html#uds-search-results) to learn more about overloading in Java.  
  
  
**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.  
  
  
**14) Can we make an abstract method final in Java?**  
No, you cannot make an abstract method final in Java because, in order to use an abstract method, you must override it but the final method cannot be overridden in Java. The compiler will flag an error if you use both abstract and final keyword together with class or method in Java. See [Core Java Volume 1 - Fundamentals](https://www.amazon.com/Core-Java-I--Fundamentals-10th/dp/0134177304?tag=javamysqlanta-20) to learn more about rules of Java programming.  
  
  
**15) Can we use non-final local variables inside an Anonymous class?**  
Yes and No, it's not allowed until Java 7, from Java 8 onward you can use a non-final local variable inside an anonymous or local inner class, provided it's effectively final, which means it's not changed after initialized. See [here](http://javarevisited.blogspot.com/2015/03/what-is-effectively-final-variable-of.html) to learn more about the effectively final concept in Java 8.  
  
  
**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. See [here](http://www.java67.com/2014/09/Why-constructor-is-important-in-java-example.html) to learn more about constructor 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. This has the potential to create subtle bugs. Read [Java Coding Guidelines](https://www.amazon.com/Java-Coding-Guidelines-Recommendations-Engineering/dp/032193315X?tag=javamysqlanta-20) to avoid such mistakes.  
  
  
**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, they are only allowed to customize individual steps which are represented by abstract method.  
  
  
**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.

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

final keyword in java is used to make any class or a method or a field as unchangeable. You can’t extend a final class, you can’t override a final method and you can’t change the value of a final field. final keyword is used to achieve high level of security while coding.

**2) What is the blank final field?**

Uninitialized final field is called blank final field.

**3) Can we change the state of an object to which a final reference variable is pointing?**

Yes, we can change the state of an object to which a final reference variable is pointing, but we can’t re-assign a new object to this final reference variable.

**4) What is the main difference between abstract methods and final methods?**

Abstract methods must be overridden in the sub classes and final methods are not at all eligible for overriding.

**5) What is the use of final class?**

A final class is very useful when you want a high level of security in your application. If you don’t want inheritance of a particular class, due to security reasons, then you can declare that class as a final.

**6) Can we change the value of an interface field? If not, why?**

No, we can’t change the value of an interface field. Because interface fields, by default, are final and static. They remain constant for whole execution of a program.

**7) Where all we can initialize a final non-static global variable if it is not initialized at the time of declaration?**

In all constructors or in any one of instance initialization blocks.

**8) What are final class, final method and final variable?**

final class —> can not be extended.

final method —> can not be overridden in the sub class.

final variable —> can not change it’s value once it is initialized.

(Click [here](https://javaconceptoftheday.com/final-keyword-in-java/) for more info on final class, final method and final variable)

**9) Where all we can initialize a final static global variable if it is not initialized at the time of declaration?**

In any one of static initialization blocks.

**10) Can we use non-final local variables inside a local inner class?**

No. Only final local variables can be used inside a local inner class.

**11) Can we declare constructors as final?**

No, constructors can not be final.

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

[**Final**](https://docs.oracle.com/javase/tutorial/java/IandI/final.html)**keyword** can be applied to **variable, method,**and **class**. Each of them has its own uses

* The **final variable** is a variable whose value **cannot** be changed at any time once assigned, it remains as a constant forever.
* The **final method cannot be overridden**
* A **final class cannot be subclassed (cannot be extended)**

### **2. What is a blank final variable?**

A **blank final variable** is a **final variable**, which is not initialized during declaration.

### **3. Can we declare final variable without initialization?**

**Yes,** We can declare a **final variable** without initialization and these final variables are called a blank final variable but must be initialized before usage.

The final variable can be initialized in the below four ways

#### ****1. Through Instance Initialization Block (IIB)****

The **Instance Initialization Block** is used to initialize the **instance data member**, this block runs every time whenever the object of the class is created. **I**nstance **I**nitialization **B**lock gets executed exactly before the code in the constructor. We can use **IIB** to initialize an **instance final variable**

public class Test

{

// Blank final variable

final int SPEED;

// Instance Initialization Block

{

SPEED = 5;

}

public static void main(String args[])

{

Test t=new Test();

System.out.println("Travelling Speed is :"+t.SPEED);

}

}

#### ****2. Through Static Initialization Block****

The **static block** is a block of code inside a Java class, which will be executed when a class is first loaded into the JVM. The **Static Initialization Block** can be used to initialize a **class final variable/static final variable**

public class Test

{

// Blank final variable

static final int SPEED;

// Static Initialization Block

static

{

SPEED = 10;

}

public static void main(String args[])

{

System.out.println("Travelling Speed is :" + SPEED);

}

}

#### ****3. Through Constructor****

A constructor also can be used to initialize a **blank final variable**.

public class Test

{

// Blank final variable

final int SPEED;

// No Parameter Constructor

Test()

{

SPEED = 15;

}

public static void main(String args[])

{

Test t=new Test();

System.out.println("Travelling Speed is :"+t.SPEED);

}

}

#### ****4. Within Method****

The **local final variable** can be initialized during declaration or any place after the declaration. It must be initialized before used.

public class Test

{

public static void main(String args[])

{

// Local final variable

final int SPEED;

SPEED = 25;

System.out.println("Travelling Speed is :" + SPEED);

}

}

### **4. What is a final method?**

When a method is declared as **final**, then it is called as a **final method,**The subclass can call the final method of the parent class but cannot **override** it.

### **5. What is a final class?**

A class declared with a **final keyword** is called a **final class**, a final class **cannot be subclassed**. This means a final class cannot be inherited by any class.

### **6. Can a main() method be declared final?**

**Yes,** the **main()** method can be declared as final and cannot be [**overridden**](https://javainterviewpoint.com/what-is-method-overriding-in-java/).

### **7. Can we declare constructor as final?**

**No,** Constructor cannot be declared as **final**. Constructors are not inherited and so it cannot be overridden, so there is no use to have a final constructor.

You will get an error like **“Illegal modifier for the constructor in type Test; only public, protected & private are permitted”**

### **8. Can we declare an interface as final?**

The sole purpose of Interface is to have the subclass implement it if we make it final it cannot be implemented. Only **public & abstract** are permitted while creating an interface

### **9. Can Final Variable be serialized in Java?**

**Yes,** the final variable can be serialized in Java

### **10. What will happen if you add final to a**List / ArrayList?

Once we created a **final List/ ArrayList** we can **add**or **remove** data from the list but the final list cannot be assigned with a value. Let’s look into the below code

import java.util.ArrayList;

import java.util.List;

public class FinalList

{

public static void main(String[] args)

{

final List myList = new ArrayList();

myList.add("one");

myList.add("two");

myList.add("three");

System.out.println(myList);

List tmp = new ArrayList();

tmp.add("four");

myList = tmp; // Throws Compilation error

myList = null;

}

}

When we assign our **myList** with **tmp** or **null**, we will be getting a [**compilation error**](https://javainterviewpoint.com/exception-handling-in-java-a-complete-guide-to-java-exceptions/) like **“The final local variable myList cannot be assigned”**

### 11. Can we make a method final in Java?

**Yes,** We can make a method final, the only constraint is that it cannot be overridden.

### **12. What is effectively final in Java?**

The term **effectively final** **variable** is introduced in **Java 8**. A variable which is not declared as **final** but the value never changed after initialization is called as **effectively final**.

### **13. Can we make the local variable be final?**

**Yes,** we can make a local variable final in Java. In fact, the **final** is the only modifier which can be applied to a local variable. If we apply any other modifier we will be getting compile time error **[only final is permitted]**.

The below code will throw compile time error as we have declared the **temp** local variable as **public**.

public class Test

{

public static void main(String[] args)

{

//Public local variable

public int temp; // Error Only final is permitted

System.out.println(temp);

}

}

### 14. Can final method be overloaded in Java?

**Yes,** the final method can be [**overloaded**](https://javainterviewpoint.com/java-method-overloading-example/) but cannot be**overridden**. Which means you can have more than one final method with the same name with different parameters.

### 15. Can we create object for final class?

**Yes,** it is possible to create an object for a final class. The best example in Java would be **String** class. The string is a final class, in almost all code we will be creating the object for it but you **cannot extend** the String class.

### **16. What is the main difference between abstract methods and final methods?**

**Abstract methods** are declared in **abstract classes** and cannot be implemented in the same class. They must be implemented in the **subclass**. The only way to use an **abstract method** is by **overriding** it

**Final methods** are quite opposite to **abstract**, final methods cannot be **overridden**.

### **17. What is the difference between abstract class and final class?**

|  |  |
| --- | --- |
| **ABSTRACT CLASS** | **FINAL CLASS** |
| **Abstract class** can be subclassed and the **abstract methods** should be overridden | **Final class** cannot be subclassed and the **final methods** cannot be overridden |
| Can contain **abstract methods** | Cannot contain **abstract methods** |
| [**Abstract class**](https://javainterviewpoint.com/abstract-class-java/) can be inherited | **Final class** cannot be inherited |
| **Abstract class** cannot be instantiated | **Final class** can be instantiated |
| **Immutable** object cannot be created | **Immutable** objects can be created |
| Not all methods of the abstract class need to have a method body (abstract methods) | All methods of the final class should have a method body |

### 18. Can we make an abstract method final in Java?

**No,** We cannot make an abstract method **final** in Java because both **abstract** and **final** are both extremes as an abstract method must be overridden while the final method cannot be overridden.

### 19. What is the difference between static and final in Java?

|  |  |  |
| --- | --- | --- |
| **STATIC KEYWORD** | **FINAL KEYWORD** |  |
| [**Static keyword**](https://javainterviewpoint.com/use-of-static-keyword-in-java/) can be applied to a **nested class, block, method**and **variables** | **Final keyword** can be applied to **class, block, method**and **variables** |  |
| We can declare **static methods** with the same signature in subclass but it is not considered as **overriding** as there won’t be any [**runtime polymorphism**](https://javainterviewpoint.com/polymorphism-java/).  If a **subclass** contains the same signature as a [**static method**](https://javainterviewpoint.com/can-we-overload-static-methods-in-java/) in the **base class**, then the method of the **subclass** **hides** the **base class** method it is called **Method Hiding**. | [**Final class**](https://javainterviewpoint.com/final-keyword-in-java-final-variable-final-method-and-final-class/) methods cannot be **overridden**. |  |
| [Static](https://javainterviewpoint.com/top-10-java-interview-questions-on-static-keyword/)**variable** can be changed after initialization | **Final variable** cannot be changed after initialization |  |
| **Static method**or **variable** can be accessed directly by the class name and doesn’t need any object as they belong to the class | Object can be created to call the final method or final variables |  |

### **20. What is a Static Final variable in Java?**

When have declared a variable as **static final** then the variable becomes a **CONSTANT**.  Only one copy of variable exists which cannot be changed by any instance.

### (1) What is final in Java?

**Java *final* is a keyword or non-access modifier which is used to restrict the user.**

### (2) Where you can use final keyword in java?

**There are many areas where we can use final keyword.**

* **It can be used with the class.**
* **It can be used with variables.**
* **It can be used with methods.**

### (3) Can we inherit final class?

**No, We cannot extend or inherit final class. If we will extend the final class, it will throw compile time error.**

### (4) Can we change the value of final variables?

**No, After initializing final variable we cannot change the value of final variable because it becomes constant.**

### (5) Can we override final method?

**No, We can't override final method in java programming language.**

### (6) Can we apply final keyword with main() method?

**Yes, we can use final keyword with main() method in java.**

### (7) What is blank final variable?

**A final variable which is not initialized at the time of declaration is known as blank final variable e.g...**

**final int a;//blank final variable**

### (8) What is static blank final variable?

**A variable which is declared with static and final keyword without any initialization is called static blank final variable e.g...**

**static final int a;//static blank final variable**

### (9) How we can initialize blank final variables?

**We can initialize blank final variables only in a constructor.**

### (10) How we can initialize static blank final variables?

**We can initialize static blank final variable in static block only.**

### (11) Can we inherit final method in sub-class?

**Yes, we can inherit final method in sub-class or child class.**

### (12) Can we declare java constructor as final?

**No, we can't apply final keyword with the constructor.**

### (13) Can we create object for final class?

**Yes, we can create object of final class but we cannot extend final class.**

### (14) Can we make interface as final?

**No, we can't make interface as final because interface must be implemented in other classes.**

### (15) Difference between abstract method and final method in java?

**Abstract method must be implemented or overridden in child class but we cannot override final method in child class or subclass.**

### (16) What is the difference between final, finally and finalize?

**There are many differences between final, finally and finalize in java.**

### (17) What is the most common predefined final class object you used in a program?

**Java String is a predefined final class object.**

### (18) Can we use non-final local variables inside a local inner class?

**No, only final local variables can be used inside a local inner class.**  
 **Visit links:**

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

* final keyword can be used with variables, methods or classes
* final variables cannot be changed, once its initialized
* final methods cannot be overridden although it can be invoked
* final classes cannot be inherited

**Q) Where all final keyword can be applied ?**

* final keyword can be applied with variables, methods or classes

**Q) What is blank final variable ?**

* A final variable declared without initializing value is known as blank final field or blank final variable

**Q) What is static blank final variable ?**

* A static final variable declared without initializing value is known as static blank final field or static blank final variable

**Q) What happens, if final variable is NOT initialized with initial value at the time of declaration ?**

* Compilation error will be thrown
* **Compile-time error:** The blank final field <final\_variable> may not have been initialized

**Q) What happens, if final variable is re-assigned in the same class body ?**

* Compilation error will be thrown
* **Compile-time error:** The final field <final\_variable> cannot be assigned

**Q) Explain order of final variable initialization in Java ?**

* The value of **final** variable can be initialized at three places (considering it’s an instance data member)
  1. Initialize where it is declared
  2. Inside instance blocks i.e.; {curly braces}
  3. Constructor
* final variable checks whether value is initialized in above order and if it doesn’t find value is initialized then compiler throws error

**Q) What happens, if final method is overridden in the sub class ?**

* Compilation error will be thrown by compiler
* **Compile-time error:** Cannot override the final method from <Parent\_Class>

**Q) Whether it is possible to invoke final methods ?**

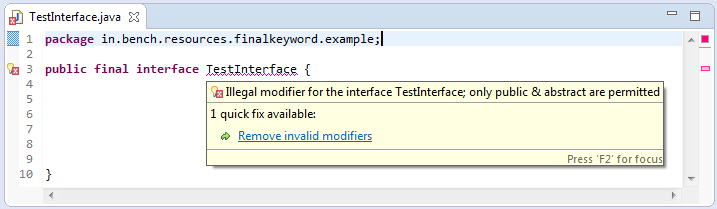
* Yes absolutely, we can very well invoke final method from inherited sub class

**Q) Whether it is possible to inherit final class ? And what happens, if we extend final class ?**

* A final class cannot be inherited
* Still, if we try to inherit then compilation error will be thrown by compiler
* **Compile-time error:** The type <Child\_Class> cannot subclass the final class <Parent\_Class>

**Q) Can we declare interface as final? If not, why explain ?**

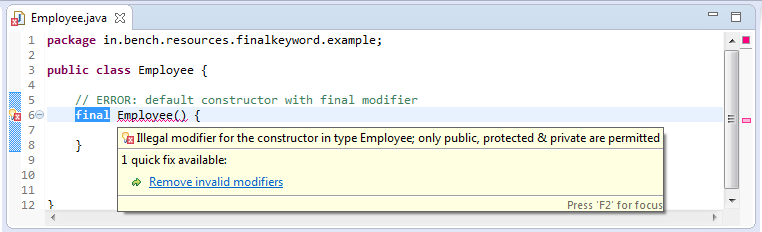
* An interface cannot be declared as final
* If we declared interface as final, then compilation error will be thrown by compiler
* **Compile-time error:** Illegal modifier for the interface TestInterface; only public & abstract are permitted

**[](http://www.benchresources.net/wp-content/uploads/2016/04/8_Interview_question_final_keyword_final_interface_error.png)**

* **Reason:** As interface contains only abstract methods, marking interface as final will restrict the interface to be inherited (and provide implementation for abstract methods)

**Q) Can we declare constructor as final ?**

* Compilation error will be thrown by marking/declaring constructor as final
* **Compile-time error:** Illegal modifier for the constructor in type Employee; only public, protected & private are permitted

**[](http://www.benchresources.net/wp-content/uploads/2016/04/9_Interview_question_final_keyword_final_constructor_error.png)**

**Q) Whether final keyword allowed in input parameters (formal parameters) of method signature ?**

* Yes it is allowed to declare final in method parameters, but final parameters cannot be changed

**Q) Whether compilation succeeds, if instance of final class is created ?**

* Yes, object of final class can be created; only it cannot be inherited

**Q) Difference between abstract method and final method ?**

* Abstract methods needs to be implemented; therefore it is overridden in the sub class
* Whereas final methods cannot be overridden in the sub class
* So conceptually final methods and abstract methods are opposite to each other where one needs to be overridden and other restrict to be overridden

**Q) Relationship between immutable class and final class ?**

* final class is a way to create immutable class in Java
* Example for final from Java library
  1. String and Math classes from java.lang package
  2. getClass(), notify(), notifyAll(), wait() are final methods from java.lang.Object class
  3. PI and Math.E are final variable in Math class

What is the use of the final keyword in Java?

The final keyword can be used with a **class, method, and variables**.

With class, it prevents inheritance by not allowing to create subclasses.

With methods, it prevents overriding, you cannot override a final method in Java.

With variable, it is treated as constant because you cannot change their value once assigned.

Can we make an array final in Java? Can you change its elements after initializing?

Yes, you can make an array final in Java and you can change it's elements as well. Both array and collection classes can be made final and you can still change their elements.

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.

Can you overload a final method in Java?

Yes, you can overload a final method in Java.

Can a static method be final in Java?

Yes, you can make a static method final in Java.

When we declare a static method as final its prevents from method hiding and encounters compile time error: Cannot override the final method from Parent.

**public** **class** **Parent** {

**final** **static** **void** **helloWorld**() {

System.out.println("Hello from Parent.");

}

}

**class** **Child** **extends** Parent {

// compile error : Cannot override the final method from Parent

**static** **void** **helloWorld**() {

System.out.println("Hello from child.");

}

}

Can a class be abstract and final?

No. It is not possible.

Can a method be abstract and final?

No, not possible. final implies that the method is complete and cannot be overridden whereas abstract marks an incomplete method and needs to be overridden.

Can you override a final method in Java?

No, you cannot override a final in Java.

Where to initialize a non-static final variable?

Non-static final variable can be initialized when declared or in constructor only. Otherwise we will encounter compilation error.

Where to initialize a static final variable in Java?

Static final variable can be initialized where it is declared or using static block.

Define a Blank final variable.

A final variable declared but not assigned is known as a blank final variable. It can be initialized within a constructor only. It raises a compilation error if it is not initialized.

What is Static blank final variable?

It is a blank final variable declared as static. That is, a final static variable declared but not given a value or not initialized is known as a static blank final variable. It can be initialized through a **static block only**.

What is Java finally block?

Java finally block is used to execute important code such as closing connection, stream etc.

It is always executed whether exception is handled or not. Java finally block follows try or catch block.

**2. What is the main difference between abstract method and final method?**

* Abstract methods must be overridden in sub class where as final methods can not be overridden in sub class

**3. What is the actual use of final class in java?**

* If a class needs some security and it should not participate in inheritance in this scenario we need to use final class.
* We can not extend final class.

**4. What will happen if we try to extend final class in java?**

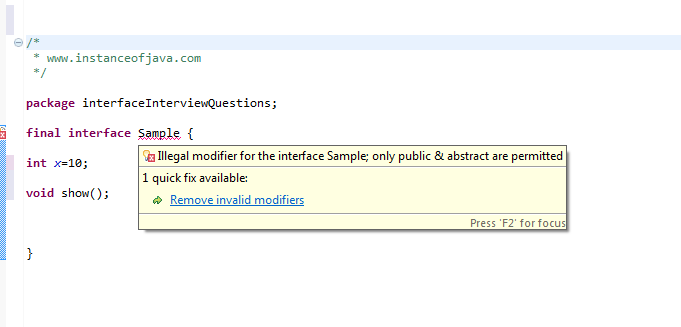
* Compile time error will come.

1. package com.finalkeywordintweviewprograms;
2. public final  Class SuperDemo{
3. int a,b;
5. public void show() {
6. System.out.println(a);
7. System.out.println(b);
9. }
10. }

1. package com.finalkeywordintweviewprograms;
2. public Class Sample  extends SuperDemo{  //The type Sample cannot subclass the final class
3. SuperDemo
4. }

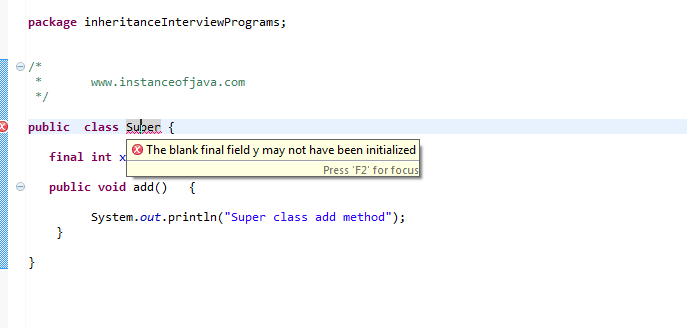
**5.Can we declare interface as final?**

* No We can not declare interface as final because interface should be implemented by some class so its not possible to declare interface as final.

[](https://3.bp.blogspot.com/-Os3PkMKXd4g/VtWhsViGPQI/AAAAAAAAAn0/e6onSIYwGrE/s1600/interface+final.png)

**6. Is it possible to declare final variables without initialization?**

* No. Its not possible to declare a final variable without initial value assigned.
* While declaring itself we need to initialize some value and that value can not be change at any time.

[](https://2.bp.blogspot.com/-f-ycwOdkLeg/VtgRQGt_JuI/AAAAAAAAAoQ/XZMEbuBWHTU/s1600/final+variable.png)

1. package com.finalkeywordintweviewprograms;
2. public final  Class Sample{
4. final int x=12,y=13;
6. public void Method() {
7. x=25;// compile time error:The final field Super.x cannot be assigned
8. y=33;// compile time error: The final field Super.y cannot be assigned
10. }
12. }

**7. Can we declare constructor as final?**

* No . Constructors can not be final.

**8.What will happen if we try to override final methods in sub classes?**

* Compile time error will come :Cannot override the final method from Super class

**9.Can we create object for final class?**

* Yes we can create object for final class.

**10.What is the most common predefined final class object you used in your code?**

* String (for example)

### Where you use final keyword in java ?

Final keyword in java are used at;

* Final at method level
* Final at class level
* Final at variable level

### What is the use of the final keyword in Java?

Final keyword can be applied to variable, method, and class. Each of them has its own uses

### What are final class, final method and final variable ?

* **Final class:**can not be extended.
* **Final method:**can not be overridden in the sub class.
* **Final class:**can not change it's value once it is initialized.

### What is Final Variable ?

The final variable is a variable whose value cannot be changed at any time once assigned, it remains as a constant forever.

### What is Final Method ?

The final method cannot be overridden

### What is Final Class ?

A final class cannot be subclassed (cannot be extended)

### What is a blank final variable ?

A blank final variable is a final variable, which is not initialized during declaration.

### What is the main difference between abstract method and final method?

Abstract methods must be overridden in sub class where as final methods can not be overridden in sub class

### When use final class in java ?

If a class needs some security and it should not participate in inheritance in this situation we need to use final class.

### What will happen if we try to extend final class in java ?

Compiler get an error

### Can we declare final variable without initialization ?

Yes, We can declare a final variable without initialization and these final variables are called a blank final variable but must be initialized before usage.

### Can we declare interface as final ?

No, because interface must be implement in other class.

### Can we declare constructor as final ?

No, Constructor cannot be declared as final. Constructors are not inherited and so it cannot be overridden, so there is no use to have a final constructor.

You will get an error like "Illegal modifier for the constructor in type Test; only public, protected & private are permitted"

### Can a main() method be declared final ?

Yes, the main() method can be declared as final and cannot be overridden.

### Can we declare constructor as final ?

No, cunstructor can't be final.

### What will happen if we try to override final methods in sub classes ?

Compile time error will come :Cannot override the final method from Super class, because you can't override final method in sub class.

### Can we create object for final class?

Yes we can create object for final class.

### Can we declare constructors as final?

No, constructors can not be final.

### Can we make a method final in Java ?

Yes, We can make a method final, the only constraint is that it cannot be overridden.

### Can we make the local variable be final ?

Yes, we can make a local variable final in Java. In fact, the final is the only modifier which can be applied to a local variable. If we apply any other modifier we will be getting compile time error [only final is permitted].

### Can we declare an interface as final ?

The sole purpose of Interface is to have the subclass implement it if we make it final it cannot be implemented. Only public & abstract are permitted while creating an interface.

### Can Final Variable be serialized in Java ?

Yes, the final variable can be serialized in Java

### What is the use of final keyword in java ?

Final keyword in java is used to make any class or a method or a field as unchangeable. We can't extend final class, not override final method, can't change value of final variable.

### Can we change the value of an interface field ?

No, we can't change the value of an interface field. Because interface fields is by default final and static. They remain constant for whole execution of a program.

### Can final method be overloaded in Java ?

Yes, the final method can be overloaded but cannot be overridden. Which means you can have more than one final method with the same name with different parameters.

### Can we create object for final class ?

Yes, it is possible to create an object for a final class. The best example in Java would be String class. The string is a final class, in almost all code we will be creating the object for it but you cannot extend the String class.

### What is the main difference between abstract methods and final methods ?

Abstract methods are declared in abstract classes and cannot be implemented in the same class. They must be implemented in the subclass. The only way to use an abstract method is by overriding it  
Final methods are quite opposite to abstract, final methods cannot be overridden.

### What is the difference between abstract class and final class ?

|  |  |  |
| --- | --- | --- |
|  | **ABSTRACT CLASS** | **FINAL CLASS** |
| 1 | Abstract class can be subclassed and the abstract methods should be overridden | Final class cannot be subclassed and the final methods cannot be overridden |
| 2 | Can contain abstract methods. | Cannot contain abstract methods. |
| 3 | Abstract class can be inherited. | Final class cannot be inherited. |
| 4 | Abstract class cannot be instantiated. | Final class can be instantiated. |
| 5 | Immutable object cannot be created. | Immutable objects can be created. |
| 6 | Not all methods of the abstract class need to have a method body (abstract methods). | All methods of the final class should have a method body. |

### Can we use non-final local variables inside a local inner class ?

No. Only final local variables can be used inside a local inner class.

### Can we change the value of an interface field? If not, why ?

No, we can’t change the value of an interface field. Because interface fields, by default, are final and static. They remain constant for whole execution of a program.

### What is the difference between static and final in Java ?

|  |  |  |
| --- | --- | --- |
|  | **ABSTRACT CLASS** | **FINAL CLASS** |
| 1 | Static keyword can be applied to a nested class, block, method and variables | Final keyword can be applied to class, block, method and variables |
| 2 | We can declare static methods with the same signature in subclass but it is not considered as overriding as there won’t be any runtime polymorphism. If a subclass contains the same signature as a static method in the base class, then the method of the subclass hides the base class method it is called Method Hiding. | Final class methods cannot be overridden. |
| 3 | Static variable can be changed after initialization | Final variable cannot be changed after initialization |
| 4 | Static method or variable can be accessed directly by the class name and doesn’t need any object as they belong to the class | Object can be created to call the final method or final variables |

### What is a Static Final variable in Java ?

When have declared a variable as static final then the variable becomes a CONSTANT. Only one copy of variable exists which cannot be changed by any instance.