# The final keyword

The final keyword is used in several different contexts as a modifier meaning that what it modifies cannot be changed in some sense.

## final classes

You will notice that a number of the classes in Java library are declared final, e.g.

public final class String

This means this class will not be subclassed, and informs the compiler that it can perform certain optimizations it otherwise could not. It also provides some benefit in regard to security and thread safety.

The compiler will not let you subclass any class that is declared final. You probably won't want or need to declare your own classes final though.

## final methods

You can also declare that methods are final. A method that is declared final cannot be overridden in a subclass. The syntax is simple, just put the keyword final after the access specifier and before the return type like this:

public final String convertCurrency()

## final fields

You may also declare fields to be final. This is not the same thing as declaring a method or class to be final. When a field is declared final, it is a constant which will not and cannot change. It can be set once (for instance when the object is constructed, but it cannot be changed after that.) Attempts to change it will generate either a compile-time error or an exception (depending on how sneaky the attempt is).

Fields that are both final, static, and public are effectively named constants. For instance a physics program might define Physics.c, the speed of light as

public class Physics {  
  
 public static final double c = 2.998E8;  
   
   
}

In the SlowCar class, the speedLimit field is likely to be both final and static though it's private.

public class SlowCar extends Car {  
  
 private final static double speedLimit = 112.65408; // kph == 70 mph  
  
 public SlowCar(String licensePlate, double speed, double maxSpeed,  
 String make, String model, int year, int numberOfPassengers, int numDoors) {  
 super(licensePlate,   
 (speed < speedLimit) ? speed : speedLimit,   
 maxSpeed, make, model, year, numberOfPassengers, numDoors);  
 }  
  
 public void accelerate(double deltaV) {  
  
 double speed = this.speed + deltaV;  
   
 if (speed > this.maxSpeed) {  
 speed = this.maxSpeed;   
 }  
   
 if (speed > speedLimit) {  
 speed = speedLimit;  
 }  
   
 if (speed < 0.0) {  
 speed = 0.0;   
 }   
   
 this.speed = speed;   
   
 }  
   
}

## final arguments

Finally, you can declare that method arguments are final. This means that the method will not directly change them. Since all arguments are passed by value, this isn't absolutely required, but it's occasionally helpful.

What can be declared final in the Car and MotorVehicle classes?

[Previous](http://docs.google.com/41.html) | [Next](http://docs.google.com/43.html) | [Top](http://docs.google.com/index.html) | [Cafe au Lait](http://www.cafeaulait.org/)

Copyright 1997-1999 Elliotte Rusty Harold

[elharo@metalab.unc.edu](mailto:elharo@metalab.unc.edu)

Last Modified February 3, 1999