**Final Keyword**

Final key word can be used with;

Variable

Method

Class

**Final Variable:**

Final variables are nothing but constants. We cannot change the value of a final variable once it is initialized.

***Example:***

class Demo{

final int MAX\_VALUE=99;

void myMethod(){

MAX\_VALUE=101;

}

public static void main(String args[]){

Demo obj=new Demo();

obj.myMethod();

}

}

**Blank final variable:**

A final variable that is not initialized at the time of declaration is known as **blank final variable**. We **must initialize the blank final variable in constructor** of the class otherwise it will throw a compilation error (Error: variable MAX\_VALUE might not have been initialized).

**Example: Final Variable not initialized**

class Demo{

//Blank final variable

final int MAX\_VALUE;

Demo(){

//It must be initialized in constructor

MAX\_VALUE=100;

}

void myMethod(){

System.out.println(MAX\_VALUE);

}

public static void main(String args[]){

Demo obj=new Demo();

obj.myMethod();

}

}

**What is use of blank final variable?**

Lets say we have a Student class which is having a field called Roll No. Since Roll No should not be changed once the student is registered, we can declare it as a final variable in a class but we cannot initialize roll no in advance for all the students(otherwise all students would be having same roll no). In such case we can declare roll no variable as blank final and we initialize this value during object

class StudentData{

//Blank final variable

final int ROLL\_NO;

StudentData(int rnum){

//It must be initialized in constructor

ROLL\_NO=rnum;

}

void myMethod(){

System.out.println("Roll no is:"+ROLL\_NO);

}

public static void main(String args[]){

StudentData obj=new StudentData(1234);

obj.myMethod();

}

}

**Uninitialized static final variable:**

A static final variable that is not initialized during declaration can only be initialized in [**static block**](https://beginnersbook.com/2013/04/java-static-class-block-methods-variables/)**.**

class Example{

//static blank final variable

static final int ROLL\_NO;

static{

ROLL\_NO=1230;

}

public static void main(String args[]){

System.out.println(Example.ROLL\_NO);

}

}

**final method:**

A final method cannot be overridden. Which means even though a sub class can call the final method of parent class without any issues but it cannot override it.

class XYZ{

final void demo(){

System.out.println("XYZ Class Method");

}

}

class ABC extends XYZ{

void demo(){

System.out.println("ABC Class Method");

}

public static void main(String args[]){

ABC obj= new ABC();

obj.demo();

}

}

The above program would throw a compilation error, however we can use the parent class final method in sub class without any issues. Lets have a look at this code: This program would run fine as we are not [overriding](https://beginnersbook.com/2014/01/method-overriding-in-java-with-example/) the final method. That shows that final methods are [inherited](https://beginnersbook.com/2013/05/java-inheritance-types/) but they are not eligible for overriding.

class XYZ{

final void demo(){

System.out.println("XYZ Class Method");

}

}

class ABC extends XYZ{

public static void main(String args[]){

ABC obj= new ABC();

obj.demo();

}

}

**final class:**

We cannot extend a final class.

final class XYZ{

}

class ABC extends XYZ{

void demo(){

System.out.println("My Method");

}

public static void main(String args[]){

ABC obj= new ABC();

obj.demo();

}

}

**Points to Remember:**  
1) A [constructor](https://beginnersbook.com/2013/03/constructors-in-java/) cannot be declared as final.  
2) Local final variable must be initializing during declaration.  
3) All variables declared in an [interface](https://beginnersbook.com/2013/05/java-interface/) are by default final.  
4) We cannot change the value of a final variable.  
5) A final method cannot be overridden.  
6) A final class not be inherited.  
7) If method parameters are declared final then the value of these parameters cannot be changed.  
8) It is a good practice to name final variable in all CAPS.  
9) final, [finally](https://beginnersbook.com/2013/04/java-finally-block/) and finalize are three different terms. finally is used in exception handling and finalize is a method that is called by JVM during [garbage collection](https://beginnersbook.com/2013/04/java-garbage-collection/).

10) Is final method inherited? Yes, final method is inherited but you cannot override it.

**Example: Final Parameters in Method:**

class FinalDemo

{

public void myMethod(int num, final String str){

// This is allowed as num is not final

num = num+10;

/\* This is not allowed as String str is final and

\* we cannot change the value of final parameter.

\* we can just use it without modifying its value.

\*/

str = str+"XYZ";

System.out.println(num+str);

}

public static void main(String args[]){

FinalDemo obj= new FinalDemo();

obj.myMethod(10, "BeginnersBook.com");

}

}