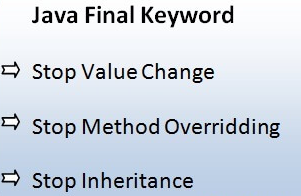
**JAVA** **final keyword**  

The **final keyword** in java is used to restrict the user. The java final keyword can be used in many context. Final can be:

1. variable
2. method
3. class

The final keyword can be applied with the variables, a final variable that have no value it is called blank final variable or uninitialized final variable. It can be initialized in the constructor only. The blank final variable can be static also which will be initialized in the static block only. We will have detailed learning of these. Let's first learn the basics of final keyword.

**1) Java final variable**

If you make any variable as final, you cannot change the value of final variable(It will be constant).

**Example**

There is a final variable speedlimit, if you want to change the value of this variable, but It can't be changed because final variable once assigned a value can never be changed.

1. **class** Bike9{
2. **final** **int** speedlimit;//final variable
3. Bike9()
4. {
5. Speedlimit=90;
6. }
7. **void** run(){
8. speedlimit=400;
9. }
10. **public** **static** **void** main(String args[]){
11. Bike9 obj=**new**  Bike9();
12. obj.run();
13. }
14. }//end of class

**2) Java final method**

If you make any method as final, you cannot override it.

**Example**

**class** Bike{

1. **final** **void** run(){System.out.println("running");} }

1. **class** Honda **extends** Bike{
2. **void** run(){System.out.println("running safely with 100kmph");}

1. **public** **static** **void** main(String args[]){
2. Honda honda= **new** Honda();
3. honda.run();
4. }
5. }

**3) Java final class**

If you make any class as final, you cannot extend it.

**Example**

1. **final** **class** Bike{}
2. **class** Honda1 **extends** Bike{
3. **void** run(){System.out.println("running safely with 100kmph");}

1. **public** **static** **void** main(String args[]){
2. Honda1 honda= **new** Honda1();
3. honda.run();
4. }

**Is final method inherited?**

Ans) Yes, final method is inherited but you cannot override it. For Example:

1. **class** Bike{
2. **final** **void** run(){System.out.println("running...");}
3. }
4. **class** Honda2 **extends** Bike{
5. **public** **static** **void** main(String args[]){
6. **new** Honda2().run();   // Honda2 h=new Honda2(); h.run();
7. }
8. }

What is blank or uninitialized final variable?

A final variable that is not initialized at the time of declaration is known as blank final variable.

If you want to create a variable that is initialized at the time of creating object and once initialized may not be changed, it is useful. For example PAN CARD number of an employee.

It can be initialized only in constructor.

Example of blank final variable

1. **class** Student{
2. **int** id;
3. String name;
4. **final** String PAN\_CARD\_NUMBER;
5. ...
6. }

**Can we initialize blank final variable?**

Yes, but only in constructor. For example:

1. **class** Bike10{
2. **final** **int** speedlimit;//blank final variable
4. Bike10(){
5. speedlimit=70;
6. System.out.println(speedlimit);
7. }
9. **public** **static** **void** main(String args[]){
10. **new** Bike10();
11. }
12. }

static blank final variable

A static final variable that is not initialized at the time of declaration is known as static blank final variable. It can be initialized only in static block.

Example of static blank final variable

1. **class** A{
2. **static** **final** **int** data;//static blank final variable
3. **static**{ data=50;}
4. **public** **static** **void** main(String args[]){
5. System.out.println(A.data);
6. }
7. }

**What is final parameter?**

If you declare any parameter as final, you cannot change the value of it.

1. **class** Bike11{
2. **int** cube(**final** **int** n){
3. n=n+2;//can't be changed as n is final
4. n\*n\*n;
5. }
6. **public** **static** **void** main(String args[]){
7. Bike11 b=**new** Bike11();
8. b.cube(5);
9. }
10. }