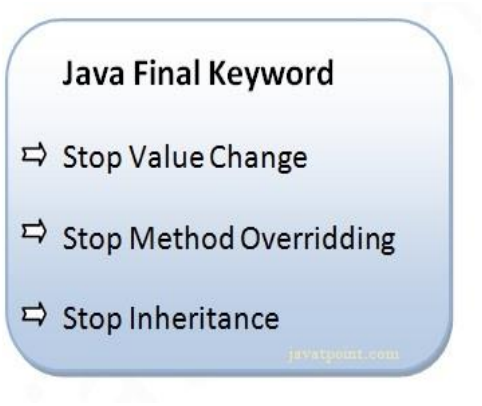
FINAL KEYWORD IN JAVA

The **final keyword** in Java is used to restrict the user. The final keyword can be used in many contexts. The final can be:

1. variable
2. method
3. class

The final keyword can be applied to the variables, a final variable that has no value is called a 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.



1) FINAL VARIABLE

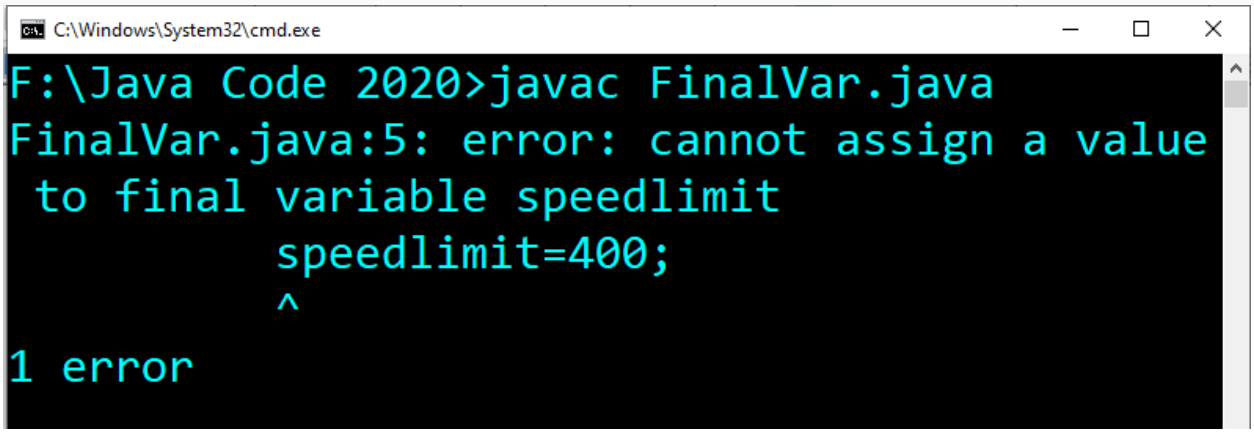
If you make any variable as final, you cannot change the value of final variable (It will be constant).

EXAMPLE OF FINAL VARIABLE

There is a final variable speed limit, we are going to change the value of this variable, but It can't be changed because final variable once assigned a value can never be changed.

```
class FinalVar{

        final int speedlimit=90;//final variable
        void run(){
         speedlimit=400;
        }
        public static void main(String args[]){
        FinalVar obj=new  FinalVar();
        obj.run();
```

```
    }
}
```



2) FINAL METHOD

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

EXAMPLE OF THE FINAL METHOD

```
class FinalMethod
{
        final void run()
        {
        System.out.println("running");
        }
}
class FinalTest extends FinalMethod {
         void run()
         {
         System.out.println("running safely with 100kmph");
         }

        public static void main(String args[]){
        FinalTest f= new FinalTest();
         f.run();
         }
       }
```

```
F:\Java Code 2020>javac FinalTest.java
FinalTest.java:10: error: run() in FinalTest
cannot override run() in FinalMethod
          void run()
               ^
  overridden method is final
1 error
```

3) FINAL CLASS

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

EXAMPLE OF FINAL CLASS

```
1.  final class Bike{}  //base
2.
3.  class Honda extends Bike{
4.    void run(){System.out.println("running safely with 100kmph");}
5.
6.    public static void main(String args[]){
7.    Honda honda= new Honda();
8.    honda.run();
9.    }
10. }
```

Output:Compile Time Error

Ques: Can we inherit the final method???

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 Honda extends Bike{
5.     public static void main(String args[]){
6.      new Honda().run();
7.     }
8.  }
```

Output:running...

STATIC KEYWORD

The **static keyword** in Java is used for memory management mainly. We can apply Java static keywords with variables, methods, blocks, and nested classes. The static keyword belongs to the class rather than an instance of the class.

The static can be:

1. variable (also known as class variable)
2. method (also known as class method)
3. block
4. nested class(we will discuss later)

JAVA STATIC VARIABLE

If you declare any variable as static, it is known static variable.

- The static variable can be used to refer to the common property of all objects (that is not unique for each object) e.g., company name of employees, college name of students, etc.
- The static variable gets memory only once in the class area at the time of class loading.

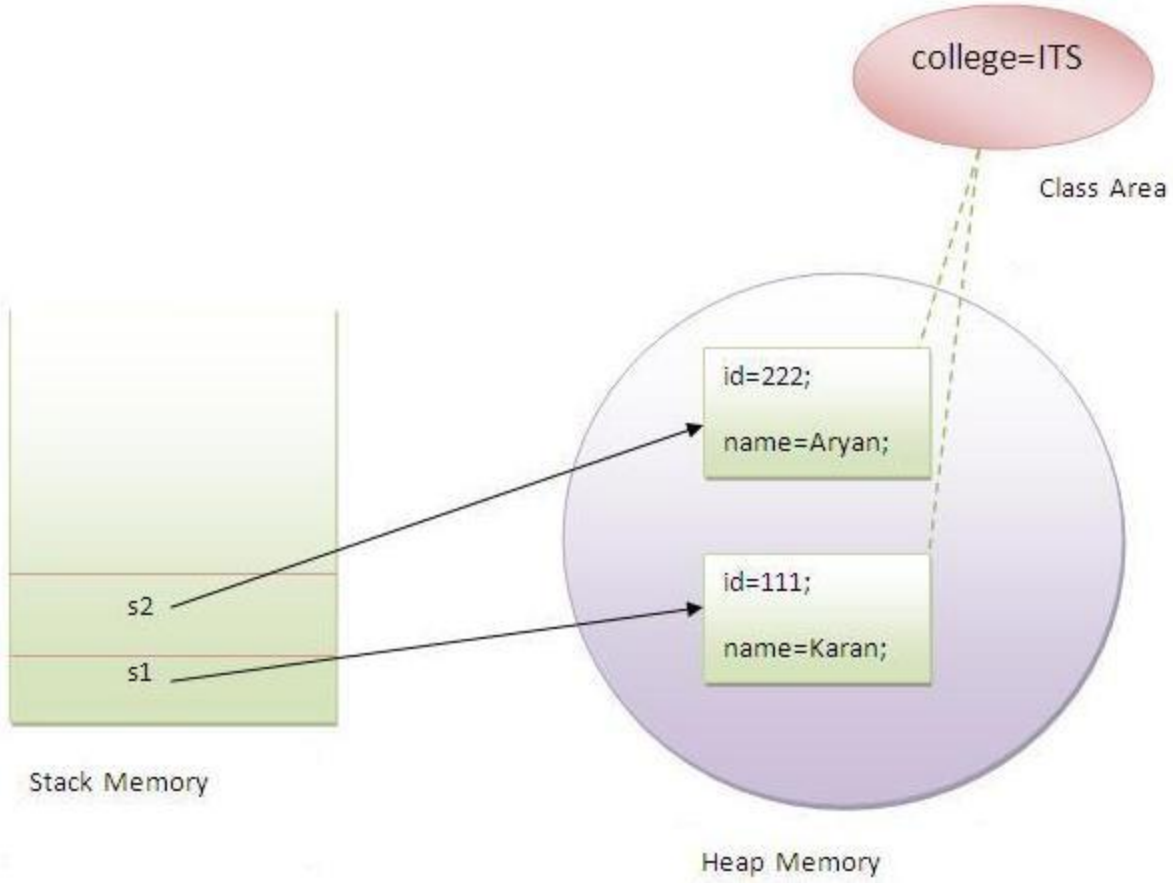
Advantage of static variable

It makes your program **memory efficient** (i.e. it saves memory).

The understanding problem without static variable

```
class Student{

 int rollno;

 String name;

static String college="ITS";

        }
```

Suppose there are 500 students in my college, now all instance data members will get memory each time when object is created. All student have its unique rollno and name so instance data member is good. Here, college refers to the common property of all objects. If we make it static, this field will get memory only once.


college=ITS
Class Area
id=222;
name=Aryan;
id=111;
name=Karan;
s2
s1
Stack Memory
Heap Memory

Example of static variable

```
1.  //Program of static variable
2.
3.  class Student{
4.     int rollno;
5.     String name;
6.     static String college ="UPES";  //in class memory
7.
8.     Student(int r,String n){
9.     rollno = r;
10.    name = n;
11.    }
12.  void display (){System.out.println(rollno+" "+name+" "+college);}
13.  public static void main(String args[]){
14.  Student s1 = new Student(111,"Karan");
15.  Student s2 = new Student(222,"Aryan");
16.  s1.display();
17.  s2.display();
18.  }
19. }
```

Output:111 Karan UPES
222 Aryan UPES

JAVA STATIC METHOD

If you apply a static keyword with any method, it is known as a static method.

- A static method belongs to the class rather than the object of a class.
- A static method can be invoked without the need for creating an instance of a class.
- static method can access static data member and can change their value of it.

//Program of changing the common property of all objects (static field).

```
class Student
{
int rollno;
String name;
static String college = "ITS";
static void change(){
college = "UPES";

  }
Student(int r, String n){
 rollno = r;
name = n;
```

```
  }
void display (){System.out.println(rollno+" "+name+" "+college);}
public static void main(String args[]){
 Student.change();

Student s1 = new Student (111,"Karan");
 Student s2 = new Student (222,"Aryan");
Student s3 = new Student (333,"Arjun");
s1.display();
 s2.display();
 s3.display();
 }
}
```

Output:

```
111 Karan UPES
222 Aryan UPES
333 Arjun UPES
```

JAVA STATIC BLOCK

- Is used to initialize the static data member.
- It is executed before main method at the time of classloading.

Syntax:

```
  static{

stsmt…..

}
```

```
1.  class A2{
2.    static
3.  {
4.  System.out.println("static block is invoked");
5.  }
6.
7.   public static void main(String args[]){
8.     System.out.println("Hello main");
9.    }
10. }
```

Output: static block is invoked
Hello main

Can we execute a program without main() method?

Ans) Yes, one of the way is static block but in previous version of JDK not in JDK 1.7.

```
1. class A3{
2.   static
3. {
4.   System.out.println("static block is invoked");
5.   System.exit(0);
6.   }
7. }
```

Output:static block is invoked (if not JDK7)

In JDK7 and above, output will be:

Output:Error: Main method not found in class A3, please define the main method as:
public static void main(String[] args)