

Difference between instance class and local variables in Java

There are lot of differences between instance variable, class variable and local variable in Java and knowing them will help you to write correct and bug free Java programs. Java is full featured programming language and provides different kind of variables like static variable also called Class variable since it belongs to whole Class, non static also called instance variable and local variables which varies in scope and value.

Thank god Java doesn't have any register variable or auto scope like C otherwise it would have so much detail to remember. static variables are common source of error in may multi-threaded java program and does require a bit of carefulness while using it. On the other hand instance variable and local variable has less sharing visibility than static variable.

Instance Variable vs Class Variable in Java



let's first see difference between instance variable and class variable also known as non static vs static variable in java. Instance variable are per instance (object) basis. If you have 5 instance of one class you will have five copies of instance variable. these are also referred as non static variable and initialized when you create instance of any object using new() operator or by using other methods like reflection e.g. `Class.newInstance()` . On the other hand Class variables are declared using static keyword and they have exact same value for every instance. static or class variable are initialized when class is first loaded into JVM memory unlike instance variable which initialized when instance is created. Static variables are similar to global variable in C and can be used to store data which is static in nature and has same value for all instance, but at same static variable also cause subtle concurrency bugs if updated by multiple threads. you can read more about static keywords in my post [secrets of static keyword in Java](#).

Instance variable vs local variable in Java

Now let's see **difference between instance variable and local variable**. local variables are local in scope and they are not visible or accessible outside their scope which is determined by {} while instance variables are visible on all parts of code based on their access modifier e.g. **public**, **private** or **protected**. only public can be accessed from outside while protected and private can be accessed from subclass and class itself. Access modifier can not be applied to local variable and you can not even make them static. only modifier which is applicable to local variable is final and only [final local variables](#) are visible inside anonymous class. value of instance variable is limited to instance, one instance can not access value of other instance in Java.

Best practices related to local, instance and static variable in Java

It's good to know some of the [best practices](#) related to declaration and use of Variables in Java while learning differences among different types of variable in Java:

1. Always **name your variable as per Java Bean naming convention**.
2. By default **give private access to your member variables** (both static and instance) and provide more access step by step e.g. from private to protected to package to public. This way you will be following encapsulation principle.
3. Always **declare local variable where you use** instead of declaring it on top of method or block.
4. **Don't hide instance or static variable** by giving same name to local variable. this may result in subtle programming bugs.
5. Be **consistent** with your variable naming convention **don't mix different convention from [different language](#)** e.g. some programming language use first word to denote type of variable e.g. bExit to denote boolean Exit variable or iNumber to denote integer Number variable. Though they are good but mixing simple names as per Java Bean naming convention with this will only lead confusion.

That's all on **difference between local vs instance vs class variable in Java**. correct understanding of their scope, value and accessibility is key to write java program.