**Java static**

Consider the class Variable. We create two instances of this class and call some methods.

**import** java.util.\*;

**class** Variable

{

 **public** **int** value;

 **public** **void** inc()

 {

 value++;

 }

 **public** **void** dec()

 {

 value--;

 }

}

**public** **class** Main

{

 **public** **static** **void** main(String[] args)

 {

 Scanner con = **new** Scanner(System.***in***);

 Variable v1 = **new** Variable();

 v1.inc();v1.inc();

 System.***out***.println(v1.value);

 Variable v2 = **new** Variable();

 v2.inc();v2.inc();v2.inc();v2.inc();

 System.***out***.println(v2.value);

 con.close();

 }

}

A ***static*** variable is common to all the instances (or objects) of the class because it is a class level variable. In other words you can say that only a single copy of static variable is created and shared among all the instances of the class. Memory allocation for such variables only happens once when the class is loaded in the memory.

 In some cases we want to have a common value for all the instances like global variable then it is much better to declare them static as this can save memory (because only single copy is created for static variables).

**import** java.util.\*;

**class** Variable

{

 **public** **static** **int** *value*;

 **public** **void** inc()

 {

 *value*++;

 }

 **public** **void** dec()

 {

 *value*--;

 }

}

**public** **class** Main

{

 **public** **static** **void** main(String[] args)

 {

 Scanner con = **new** Scanner(System.***in***);

 Variable v1 = **new** Variable();

 v1.inc();v1.inc();

 // static variables must be accessed in a static way

 System.***out***.println(Variable.*value*);

 Variable v2 = **new** Variable();

 v2.inc();v2.inc();v2.inc();v2.inc();

 System.***out***.println(Variable.*value*);

 con.close();

 }

}

**Static variable initialization**

* Static variables are initialized when class is loaded.
* Static variables are initialized before any object of that class is created.
* Static variables are initialized before any static method of the class executes.

Static block that gets executed exactly once, when the class is first loaded.

**import** java.util.\*;

**class** Variable

{

 **public** **static** **int** *value* = 10;

 **public** **void** inc()

 {

 *value*++;

 }

 **public** **void** dec()

 {

 *value*--;

 }

}

**public** **class** Main

{

 **public** **static** **void** main(String[] args)

 {

 Scanner con = **new** Scanner(System.***in***);

 Variable v1 = **new** Variable();

 v1.inc();

 System.***out***.println(Variable.*value*);

 Variable v2 = **new** Variable();

 v2.inc();

 Variable.*value*++;

 System.***out***.println(Variable.*value*);

 Variable v3 = **new** Variable();

 v3.inc();v3.inc();

 Variable.*value*++;

 System.***out***.println(Variable.*value*);

 con.close();

 }

}