*Static keywords*

It means single copy storage. It can be applied to variable, method, inner class and static block. Purpose of static keyword is memory management (it saves memory).

Static Keyword Applicable for

1. *Variable*

This is also known as class variables and variable with static keyword called as “static variable.”

It is used to refer the common property of all the objects.

Static variables get loaded into memory at the time of class loading.

***Way to declaration and initialization of static variable***

**static** **int** *a*; //Declaration of static variable

**static** **int** *b*=5; //Initialization of static variable

***way to access the static variables***

1. By using class name
2. By using object name

**package** com.velocity;

**public** **class** Student {

**static** **int** *a* = 5;

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

System.***out***.println("value of a>>" + *a*);// in same class only we can call directly

System.***out***.println("By class name value of a>>" + Student.*a*); // by using class name

Student student = **new** Student();// creating the object

System.***out***.println("By object name value of a>>" + student.*a*); // by using object name

}

}

Output

value of a>>5

By class name value of a>>5

By object name value of a>>5

***Example 2 if static variable in another class***

**package** com.velocity;

**public** **class** Demo {

**static** **int** *a* = 5; // static variable in other class

}

**package** com.velocity;

**public** **class** Student {

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

System.***out***.println("By class name value of a>>" + Demo.*a*); // by using class name

Demo demo = **new** Demo();// creating the object

System.***out***.println("By object name value of a>>" + demo.*a*); // by using object name

}

}

Output

By class name value of a>>5

By object name value of a>>5

Purpose of static keyword is memory management (it saves memory).let see the below example

**package** com.velocity;

**public** **class** Student {

**int** a = 10;

**static** **int** *b* = 10;

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

Student student = **new** Student();

System.***out***.println("non static a>>" + student.a++);

System.***out***.println("static b>>" + student.*b*++);

Student student1 = **new** Student();

System.***out***.println("non static a>>" + student1.a++);

System.***out***.println("static b>>" + student1.*b*++);

Student student2 = **new** Student();

System.***out***.println("non static a>>" + student2.a++);

System.***out***.println("static b>>" + student2.*b*++);

}

}

Output

non static a>>10

static b>>10

non static a>>10

static b>>11

non static a>>10

static b>>12

***2. static method***

If you define any method with static keyword then it is called as static method.

It loads into memory before object creation.

It can access only static data member only.

It is known as class method.

It belongs to class rather than object of class.

**package** com.velocity;

**public** **class** Student {

**static** **void** x1() {

System.***out***.println("This is static method");

}

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

Student.*x1*();//by class name

Student student=**new** Student();

student.*x1*();//by object creation

}

}

Output

This is static method

This is static method

***3.Static block***

It is group of statements that are executed when class is loading into memory by Class loader.

It is executed before the main method at the time of classloading

It is widely used to create the static resource.

We cannot access non-static variable into static block.

Example>>

**package** com.velocity;

**public** **class** Student {

**static** {

System.***out***.println("this is the static block");

}

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

System.***out***.println("this is main method");

}

}

Output

this is the static block

this is main method

*point to be remember*

1. Local variables cannot be static.
2. We cannot call non-static member from static member because static variables stored into memory before object creation and non-static variables stored into memory after object creation.
3. Constructor cannot be static.
4. Main method is static method.