Static variables:

- If the value of a variable is not varied from object to object such type of variables is not recommended to declare as instance variables. We have to declare such type of variables at class level by using static modifier.
- In the case of instance variables for every object a separate copy will be created but in the case of static variables for entire class only one copy will be created and shared by every object of that class.
- Static variables will be created at the time of class loading and destroyed at the time of class unloading hence the scope of the static variable is exactly same as the scope of the .class file.
- Static variables will be stored in method area. Static variables should be declared with in the class directly but outside of any method or block or constructor.
- Static variables can be accessed from both instance and static areas directly.
- We can access static variables either by class name or by object reference but usage of class name is recommended.
- But within the same class it is not required to use class name we can access directly.
- Static variables also known as class level variables or fields.

Example:

```
class Test{
        int x=100;
        static int y=200;

        public static void main(String[] args){
                Test t1=new Test();
                t1.x=888;
                t1.y=999;
                Test t2=new Test();
                System.out.println(t2.x+"----"+t2.y);//100------- 999
        }
}
```

How to access static variable.

There are two ways to access the static variables.

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

Program for static variables

```
class Statickeyword{
    public static void main(String args[]){
    Statickeyword obj= new Statickeyword();
    static int a=5;
    System.out.println(Statickeyword.a); //by using class name
    System.out.println(obj.a);  // by using object name
 }
}
```

Output-

```
5

5
```

In this example, we are calling static variable by using statickeyword class name, also calling it by using object obj as shown in above program.

Why it is called as single copy storage?

Because for multiple object only one copy of variable will be made for a given static variable.

```
package com.velocity;

public class Test10 {

	int a = 5;
	static int b = 5;

	public static void main(String args[]) {

		Test10 sd2 = new Test10();
		System.out.println("non static>>"+sd2.a++);
		System.out.println("static>>"+sd2.b++);
		Test10 sd3 = new Test10();
		System.out.println("non static>>"+sd3.a++);
		System.out.println("static>>"+sd3.b++);
		Test10 sd4 = new Test10();
		System.out.println("non static>>"+sd4.a++);
		System.out.println("static>>"+sd4.b++);
		Test10 sd5 = new Test10();
		System.out.println("non static>>"+sd5.a++);
```

```
        System.out.println("static>>"+sd5.b++);

    } }
```

Output-

```
non static>>5
static>>5
non static>>5
static>>6
non static>>5
static>>7
non static>>5
static>>8
```

Note- 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.

How to access static members from non-static members. Following program shows you.

```
class StaticDemo{
        void test(){
                System.out.println("This is non static method");
                StaticDemo().x1 () //calling static method from non static method.
        }

        static void x1(){
                System.out.println("This is static method");
        }

        public static void main(String args[]){
                StaticDemo s= new StaticDemo();
        s.test();
        }
```

Output-

This is non static method.

This is static method.

Static method-

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

It belongs to class rather than object of class.

It loads into memory before object creation.

It can access only static data member only.

Note: - Main method is static method.

```
class StaticDemo{
	static void x1(){
		System.out.println("This is static method.");
	}

	public static void main(String args[]){
		StaticDemo.x1();
	}
}
```

Output-

This is static method.

Static block-

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

It is widely used to create the static resource.

We cannot access non-static variable into static block.

It is always executed first.

```
package com.test;

public class Test3 {

    static {
        System.out.println("this is the 1st static 
block...");
    }
    public static void main(String[] args) {

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

Output-

```
this is the 1st static block...
this is main method..
```

Here, we will get first output is "This is 1st static block" because it is executed first than main method.

Note- Outer class cannot be static but inner class can be static.

Constructor cannot be static.

Local variables cannot be static