**Static** :

When a member is declared static, it can be accessed before any objects of its class are created. So we can use static variable and method without an object.

1. Variable
2. Method

**Static Variable** : Declared **outside** of any method but **inside a class**. A static variable can be accessed directly by the class name and **doesn’t need any object** to access. We **can not** declare static variable inside static method, we have to declare in class not in method.

The static variable can be used to refer the common property of all objects (that is not unique for each object) e.g. company name of employees, college name of students etc.

**Static Method** : We can’t use non-static variable in static method, Static method can access only static variables. A static method can be accessed directly by the class name and **doesn’t need any object**. A static method can call only other static methods and cannot call a non-static method from it. If we want to call a non-static method from a static method, we need to create an object of class, with help of object we can call non-static method from static method.

One static method can call other static method of same class without class name in same class ONLY.

**Note:** main method is static, since it must be accessible for an application to run, before any instantiation takes place.

**Explanation** : Open Example from eclipse, write it on board. If I want to use a method written in a class I need to create an object of a class and then with help of object I can access a method of that class. Same for a variable of a class.

But, If I made that method as a static, then no need to create an object of that class.

We can directly call that method as class\_name.method name(); same applies for class variable, no need to create class object to access static class variable. Use class name.variable name.

**Static method overloading and overriding** :

Static method can be overloaded. Static method overriding is nothing but method hiding.