Structure of Java Program:

Static variables

- If the value of a variable is not varied from object to object such type of variables is not recommended to be declared as instance variables.
- We have to declare such types of variables at class level by using static modifiers.
- In the case of instance variables for every object a separate copy will be created but in the case of static variables for the 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 the same as the scope of the .class file.
- Static variables will be stored in the method area. Static variables should be declared within 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 names we can access directly.

We can access static variables in 2 ways

1. Using className
2. Using reference variables

Note: Static variables also known as class level variables or fields.

Static methods

Methods which are available at the class level are referred to as “static methods”.
These methods are referred to as utility methods.
Inside the static methods we can access only static variables.
If we try to access the instance variables directly then it would result in **“CompileTimeError”.**

static block

- These are the blocks which gets executed automatically at the loading the .class files
- If we want to perform any activity at the time of loading a .class file we have to define that activity inside the static block.
- We can write any no of static blocks, those static blocks will be executed from top to bottom.
- Normally a static block is used to perform initialization of the static variables.

Difference with respect static and non static members of a class static

- These variables are called “class variables”.
- These variables will get memory in the method area.
- If the value does not change from object to object then we need to use static variables.
- Inside a static area we can access static variables only.
- Static variables are created using static keywords.

Non-static

- These variables are called “instance variables”.
- These variables will get memory in the heap area.
- If the value changes from object to object then we need to use “non-static” variables.
- Inside a nonstatic area we can access both static and non-static variables.
- Non-static variables are created without using the “static” keyword.