The static keyword in Java is used to indicate that a member (variable or method) belongs to the class itself rather than to any specific instance of the class. Here’s a brief overview:

1. **Static Variables:** A static variable is shared among all instances of a class.

class MyClass {

static int sharedCounter = 0;

// All instances of MyClass will share this counter.

}

//You can use this variable without creating any object of the class MyClass

MyClass.sharedCounter;

1. **Static Methods:** A static method can be called without creating an instance of the class.

class MyClass {

static void display() {

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

}

}

MyClass.display(); // Calling static method without creating an object

1. **Static Blocks:** A static block is used for static initialization of a class. It gets executed when the class is loaded.

class MyClass {

static {

System.out.println("Static block executed"); // This will be printed once when the class is loaded

}

}

\*\*\*\*Using static block in public class then we don’t need to do anything but if it is another class we must trigger it.

You can do that by creating object of that class or do something like accessing static variables.

1. **Static Classes (Nested):** You can declare a static nested class inside another class.

class OuterClass {

static class InnerClass {

// This is a static nested class.

}

}

**Creating Instances:** Since the inner class is static, you can create instances of it without an instance of the outer class.

OuterClass.InnerClass innerObject = new OuterClass.InnerClass();

innerObject.display(); // Calls the method in the static inner class

A static inner class is a nested class that is declared static within another class. Because it’s static, it doesn’t need an instance of the outer class to be instantiated.

The static inner class can access the static members of the outer class directly, but it cannot access non-static members unless it has an instance of the outer class.

class OuterClass {

static int outerStaticVar = 10;

static class InnerClass {

void display() {

System.out.println("Outer static variable: " + outerStaticVar);

}

}

public static void main(String[] args) {

OuterClass.InnerClass innerObject = new OuterClass.InnerClass();

innerObject.display(); // Outputs: Outer static variable: 10

}

}

A black background with white text

Description automatically generated

