Static vs Non-Static in Java

Introduction

In Java, the 'static' keyword is used to indicate that a particular member (variable or method) belongs to the class rather than to any specific instance of the class.

Static vs Non-Static Comparison

1. Static Variables:
   - Belong to the class, not to any one instance.
   - Shared among all objects of the class.
   - Initialized only once at the time of class loading.
   - Can be accessed using ClassName.variable or through objects.

2. Non-Static Variables:
   - Each object has its own copy.
   - Initialized when the object is created.
   - Accessed through object references.

3. Static Methods:
   - Can be called without creating an instance.
   - Can access only static data directly.
   - Cannot use 'this' or 'super' keyword.

4. Non-Static Methods:
   - Belong to the object.
   - Can access both static and non-static data.
   - Can use 'this' and 'super'.

5. Static Block:
   - Runs once when the class is loaded.
   - Used to initialize static variables.

6. Use Case of Static:
   - Utility or helper methods (e.g., Math.sqrt())
   - Constants (e.g., final static int MAX = 100)
   - Keeping track of object count across instances

7. Example:

```
class Demo {
    static int count = 0;
    int id;
    Demo() {
        count++;
        id = count;
    }
}

Demo a = new Demo(); // a.id = 1
Demo b = new Demo(); // b.id = 2
// count is shared: Demo.count = 2
```