**Final:**

* Final variable can’t change is value or reference, it basically functions as a constant.
* Final Method can’t be **overridden**, but can be overloaded
* Final class can’t be **inherited** by any other class.

**Static:**

* Static variables only get initialized once at the **start of execution**; it belongs to the class not object. One copy is shared between all instance.
* Static methods can’t be **overridden**.
* We can’t use static on a **top-level class**. But we can have **nested static class**. In this case that nested static class can only access static member of its parent class.

**Abstract:**

* Abstract class can’t be **instantiated**.
* Abstract class can contain 0 or more **abstract methods**.
* Abstract class can’t be **final** since they are fundamentally opposite. (Abstract class must be inherited and final class can’t be extended)
* Abstract method doesn’t have a body, if any class have abstract method, it must be declared as abstract class
* Abstract class can implement interface, without overriding all (or any) methods of interface. In this case, the class implementing the abstract class will have to provide implementation of the interface methods

**Transient:** Transient variable is not serialized (process of converting object into byte stream). So, when the object is deserialized the transient variables value will be the default instead of the value we assigned. These variables changes are not saved in persistent memory.

Reference: https://www.baeldung.com/java-transient-keyword

**Volatile:** It tells compiler the value of the variable must never be cached as its value can change outside of the program scope. It is another way (other ways: synchronized, atomic etc.) of making class thread safe.