Java 'static' Keyword Notes

Step-by-Step Notes on the 'static' Keyword in Java:

1. Static Variables:

- Belong to the class, not to any specific instance.
- Shared among all instances of the class.
- Initialized only once, at the time of class loading.

Example:

```java
public class StaticVariableExample {
    static int count = 0;
    StaticVariableExample() {
        count++;
    }
}
```

2. Static Methods:

- Can be called without creating an object of the class.
- Can only access static variables or call other static methods directly.
- Cannot access instance variables or methods unless through an object.

Example:

```java
public class StaticMethodExample {
    static void displayMessage() {
        System.out.println("Static method called.");
```

```
    }
}
```

3. Static Blocks:

- Used to initialize static variables or execute code at the time of class loading.
- Executes only once, in the order they appear in the class.

Example:

```java
public class StaticBlockExample {
    static int data;
    static {
        data = 10;
        System.out.println("Static block executed.");
    }
}
```

4. Static Classes (Nested Static Classes):

- A nested class marked with `static` does not require an outer class instance.
- Can access only static members of the outer class directly.

Example:

```java
public class OuterClass {
    static class NestedStaticClass {
        void display() {
            System.out.println("Static nested class.");
```

```
        }
    }
}
```

5. Restrictions of `static`:

   - Cannot use `this` or `super` in a static context.

   - Cannot override static methods (hiding occurs instead).

Summary Table:

Use Case	Behavior
Static Variable	Shared across objects.
Static Method	No instance required.
Static Block	Executes once.
Static Class	No outer instance.