In Java, the `final` keyword is used to define constants, prevent method overriding, and inheritance. It can be applied to variables, methods, and classes, and it has specific meanings in each context.

1. Final Variables

When you declare a variable as `final`, its value cannot be changed once it has been assigned. This makes the variable a constant.

```
java
Copy code
public class FinalVariableExample {
    public static void main(String[] args) {
        final int MAX_VALUE = 100;
        System.out.println("The maximum value is: " + MAX_VALUE);

        // MAX_VALUE = 200; // This will cause a compilation error because
MAX_VALUE is final
    }
}
```

2. Final Methods

A method declared as `final` cannot be overridden by subclasses. This is useful when you want to ensure that the method's implementation remains unchanged.

```
java
Copy code
class Parent {
    public final void display() {
        System.out.println("Display method in Parent class.");
    }
}

class Child extends Parent {
    // This will cause a compilation error
    // @Override
    // public void display() {
    //     System.out.println("Display method in Child class.");
    // }
}

public class FinalMethodExample {
    public static void main(String[] args) {
        Parent p = new Parent();
        p.display(); // Calls the final method
    }
}
```

3. Final Classes

A class declared as `final` cannot be subclassed. This is useful when you want to prevent inheritance and ensure the class's behavior remains unchanged.

```
java
Copy code
final class FinalClass {
    public void show() {
        System.out.println("This is a final class.");
    }
}

// This will cause a compilation error
// class SubClass extends FinalClass {
// }

public class FinalClassExample {
    public static void main(String[] args) {
        FinalClass fc = new FinalClass();
        fc.show(); // Calls the method from the final class
    }
}
```

Summary

- **Final Variables**: Constants that cannot be changed once initialized.
- **Final Methods**: Methods that cannot be overridden in subclasses.
- **Final Classes**: Classes that cannot be extended or subclassed.