**Poor Style: Non-final Public Static Field Development Mitigation SOP**

Typically, you do not want to provide external classes direct access to your object's member fields since a public field can be changed by any external class. Good object oriented designed uses encapsulation to prevent implementation details, such as member fields, from being exposed to other classes. Further, if the system assumes that this field cannot be changed, then malicious code might be able to adversely change the behavior of the system.

**Defense Against Poor Style: Non-final Public Static Field**

If you intend to expose a field as a constant value, the field should be declared as public static final, otherwise declare the field private.

**Example**

In the following code, the field ERROR\_CODE is declared as public and static, but not final:

public class MyClass

{

public static int ERROR\_CODE = 100;

//...

}

In this case, malicious code might be able to change this error code and cause the program to behave in an unexpected manner.

**References**

1. [HP Enterprise Security - Poor Style: Non-final Public Static Field](http://www.hpenterprisesecurity.com/vulncat/en/vulncat/java/poor_style_non-final_public_static_field.html)
2. [CWE - Definition 500](https://cwe.mitre.org/data/definitions/500.html)