Writing constants in code

Generally, it is bad practice to write constants (i.e., numbers) directly in your code – especially when you use them more than once.

This is because it is very difficult to find them all later if you need to change them.

Instead define them as static final constants. They can be private if not in the API.

In fact in a large software package, you may declare a utility class with only the constants that are defined throughout the code.

Static or non-static?

A constant can be either static or non-static … usually best practice is to make them static, but for a private constant it doesn’t matter much.

It only really matters if:

* you need the constant to exist without any instances being created
* you want to access the constant from a static method

Employee c,d;

// …

c == d // test whether references are the same

c.equals(d) // test whether contents are the same

in equals(), you need to check to see if you can cast the object o to whatever class you are using

* test whether it is an instance of the class (instanceof)
* test that the pointer is not null (o != null)

If either are false, then the objects can’t be equal