Class Design Guidelines – Clarity

When all the class design guidelines come together, in combination they provide excellent clarity. When everything is running smoothly within the program and everything is in working order, your program will maintain a very nice clarity to it. A clear, organized explanation of your program is a good sign that it possesses clarity. The less restrictions and impositions that your classes place on each other the better clarity you will achieve. When a program is convoluted and filled with classes and methods that are enveloping each other and intertwining, the clarity of the program will decline heavily. Methods need to be defined intuitively without causing confusion. Data fields should not be declared when they can be derived from other data fields. There are many good practices that one can perform in order to improve clarity within a program. One practice I find useful is using enums instead of Boolean where appropriate, because it can bring a lot of clarity into your program. One of the things we have been going over heavily and are even using in this assignment is enums. Enums is a special class that represents a group of constants. In the English language, a word can be spelled the same and have several different meanings. For example, “plane” could be referencing an airplane or it could be referencing an abstract level of existence (amongst other definitions but I’ll focus on these two). To avoid this in java, using enums defines that variable to a single definition, or value. If we defined plane as an airplane, then the 2nd definition would no longer exist within that enums class.

Take for example: (taken from <https://www.david-merrick.com/2017/11/28/good-java-practices-for-clarity/>)

public static class AccessControl {

public enum Level {

PUBLIC, PRIVATE

};

private Level access;

public Level getAccessLevel(){

return access;

}

}

This program uses enums to set a specific value for public and private. Private will not grant access whereas public will grant access. These values are forever instilled throughout all the points in the program.