Lab002 Part 2 // Anh-Thy Ho

Louden and Lambert’s criteria contains efficiency, regularity, security and extensibility. Java is more efficient, in my opinion, as there are more defined ways it can be used and there are existing expressions which can be implemented. For example, to create something like a maximum number function in Scheme, there needs to be recursion. However, in Java, the iteration of a list of numbers can be done within a loop. Also, Java does not require the strict syntax that Scheme does with the extensive parenthesis or not being able to redefine variables. In this sense, Java is a more user-friendly and a language that is easier to manipulate over time.

Talking about regularity, both languages follow conventions and are relatively structured. For example, Java has its own notations such as how to create a function, name variables, and create something like a loop. Scheme has a similar structure where functions are defined and there are usages such as let and lambda. However, Java has more regularity in that there can be specifically define variables carried over multiple times.

Security of both in terms of what can be called into error and checking, they are both secure. Java and Scheme will both have compilers check for errors. When there are simple syntax errors, they are picked up and easily accessible. In Scheme, the security is even stronger because sometimes the error will run out of time and stop itself. In Java, sometimes the program will run until it is manually stopped. I don’t know enough about extensibility to compare Java and Scheme but Scheme does have the abilities to create macros. Java have plenty of classes which can be added to a program to implement function such as ServiceLoader or methods like iterator.