This module was pretty interesting to me. We finally start to take off our training wheels and make our own Java classes instead of relying on the ones in the *bookClassses* directory. Learning how to create our own classes gave me a wider understanding of how to organize our code in a strategic and logical way that is easy to understand, maintainable and secure, and also gave me a vague understanding of how programming works.

Understanding the practical purpose of method *visibility* widened my perspective of how to organize and secure my code, not just in Java, but in programming entirely. Before reading this module, I am quite familiar with the concept of making a Class as it is very similar to *struct*, as in structure, in C programming language. However, the concept of adding a visibility on a method is new to me, thus I did not realize the practical use for it until I read about it in Module 6. Though I am still not on a deeper level of understanding to fully grasp the contrast between *struct* in C and *Class* in Java other than the obvious ones like C does not have method visibility just like Java has. But since, Java is an object-oriented language, it gave me the impression that its only fitting to have visibility in every method since, in Java, from what I understand, everything revolves around classes.

Furthermore, leaning the implementations of Java Classes deepen my perspective of how programming really works. So far, programming to me has just been all about fundamentals and core concepts, and understanding how things work under the hood. Moreover, it occurs to me that in order to be effective in programming, one must have a structure that is prevalent, or in other words, similar to that of how we organize our life in general. For instance, Java Classes are similar to cooking a meal where it requires ingredients (fields in Java class), recipe (constructors), procedures (methods), and we can’t really mix French fry with soy sauce (visibility). Or can we? I know it sounds too simplistic, and some may disagree with my comparison, but deriving complex programming solutions from real world objects can make code organization a lot easier and readable. Also, it can alleviate the pain of deciding what to name a variable!

In conclusion, I have learned how to create our own Java Classes that encapsulates everything related to the Class object. It is a very effective way of organizing your code in an object-oriented way. Also, I have learned that programming, in general, is not about the language, but the underlying fundamentals of programming and how you organize, structure, and implement your code in a practical and logical manner is what really counts.j