Not too much new knowledge this week, I think this week is like remembering the lessons that were taught when I was in college, it’s just using Ruby. It doesn't feel boring even though I have to repeat the lessons I've learned, because the instructors provide many case studies and try to engage the participants to actively participate in solving a problem.

At the first meeting I learned about the difference between procedural programming and OOP. Both have their respective advantages and have the same use, both can solve programming problems, but here I learned that procedural programming is action-oriented while OOP is object-oriented.

Before I learned about OOP in the first week meeting of the Intermediate GIGIH Generation program I only understood the concept of OOP and how OOP works, but now I have a new perspective on how OOP can solve problems both large and small by creating multiple entities from one main keyword. For example, the case study that was studied last week was about games. The first step I have to do is find out what classes I will need to meet a game's criteria. The main name of this class must be a keyword from the theme of the problem itself, namely 'Game'.

This week I also learned about inheritance and polymorphism in OOP. Actually, I already got this material when I was in college. Here I re-learn about inheritance where I used to learn inheritance from PHP and Java programming. I realize that every language has some differences in implementing OOP. Java is famous for its complexity to write 1 for classes and their components. PHP is almost like Java but more simplified. Because in PHP, I found the Trait mechanism, the concept is similar to inheritance in Ruby, except that Trait allows a class to have more than one parent.

What I still don't understand is when is the right time to use the concept of composition. Is composition only used on main classes which require multiple classes, so it’s need to use composition instead of inheritance? Is it not possible to use the concept of composition in dependent classes? Looks like I'll have to do a different case study to find the answer to the question.