In this module, we reinforced our knowledge about Java loops and kicked it up a notch with nested loops where additional loops are executed within the scope of another loop. Loops within loops. Since I had previous experience with coding, loop has already second nature to me and it made it really easy to grasp the concept of using loops to do awesome program such as manipulating each pixel of an image. But in spite of my previous experience with coding, I still had few learning moments in this module where I got stuck even with the most fundamental mistakes and also some aha moments.

While I was working on our Simple Collage homework, I anticipated it to be just like the Tri-effect in module 3 where the picture object is to be divided into three parts, each having a filter being applied to, but split vertically this time instead of horizontal. Normally it would be easy since I’ve done something similar before, but right when I finished coding the barebones of program, I got stuck for an hour. I have tried tracing back the execution, reading each line to find something I must have missed, and even “rubber ducked” it where I try to explain how my code works to the action figure on my desk. Also, I threw some print statements around too hoping that I catch the bug in my code and found out that the program I made wasn’t even executing the loop in the first place. The error message is not helpful at all for me to find the bug. Right when I’m about to lose it, I notice something hiding in plain sight, THAT DAMN SEMI-COLON after the for-loop controls! “for (int x = 0; x < picHeight; x++);”. The semi colon right next to the closing parenthesis got me really good. Since semi-colon in Java is the syntax for ending a statement, you should never end a for-loop before it even executes the loop code block.

In conclusion, coding is sometimes annoying where you find yourself working on a project and all of a sudden it throws an unexpected error right at you, even when you are certain that it should work. Due to this embarrassing blunder, I confirm that fundamentals are extremely important.