I anticipated that creating the methods would be the most difficult for me. However, I specifically struggled with three areas of the lab: creating unique, successive ID numbers for each Employee object, resetting the hours after the getWeeklyCheck() method is initialized, and getting the Employee object to create a PayCheck object. Most of this boiled down to my unfamiliarity with the Java syntax. For getWeeklyCheck(), I realized I needed to store the Paycheck object in a named variable, reset the Employee object's hours, and then return the stored Paycheck object via the variable. In order to create the PayCheck object, I simply needed to pass the Employee object's attributes as parameters in creating the Paycheck. The ID problem ended up being the most time-consuming, and so it helped me to reference the Canvas modules and solutions on forums like Stack Exchange (linked for reference). There may have been a better way of doing this, but I wasn't able to get this to work when I put similar code in the constructor.

I definitely am more comfortable using the Java syntax. Creating methods, constructors, and objects feel more familiar to me, and I have a better understanding of the class/object relationship. Additionally, I enjoy using this IDE and already prefer it to Visual Studio Code. However, there are still three things I feel confused about: the ID generator and the delta parameter. For the ID generator, I'm not sure why you start the ID constructor at 1 since the ID should increase by 1 every time the generateNewId() method is run. I assumed it would start at 0, then generate an initial ID of 1, but that didn't seem to work. Additionally, I understand that the delta parameter basically gives the assertEquals() tests a margin of error due to differences in how the computer rounds numbers, but if you're literally feeding it the same two numbers in the test, why does it still require the delta? Finally, when Employee.getWeeklyCheck() runs, what is the actual name of the PayCheck object we create? Would it be something like Employee.Paycheck.thisWeeksPay?