Collin Heretick
R#00806673

Final Project

I began this project the same way I begin every assignment, by adding it to my GitHub. This was mainly to ensure that I would not lose any progress. Additionally, I created a text file to record my progress and to plan out my project. One of the first things I needed to do was to decide on what kind of data structure would be most appropriate for this project. I decided that a two-dimensional array would be the simplest way to store the data because I could easily separate the salaries by professor while maintaining scalability and providing a simple way to retrieve the desired data. Rather than jumping straight into writing the actual methods for the "ProfessorsSalary" class, I opted to start with the main method. By doing this, I could rely on a well-defined outline which would keep me on track and organize what the program actually needed to accomplish.

Before I could begin working on the class's methods, I had to create a constructor which would populate the class's array with random salaries based on two parameters, the number of years and the number of professors. After making a constructor method, I decided to make a method which would print out the array containing the professors' salaries, which would make developing the rest of the methods much easier because I would have a way of visualizing and checking the data being used. Now that these more preliminary structures were in place, I could finally start coding the rest of the methods. Because none of the methods relied on each other, other than the constructor method, it made sense to write them in order of appearance in the main method. This way, it would be easier to remain organized.

After implementing a few of the methods, I realized that the print method I had made earlier on was not displaying all ten years of salaries and instead was only displaying five. The first thing I did after realizing this was to make sure that the constructor was actually populating ten years' worth of salaries, using the debugging tool, which it was. After taking a closer look at the constructor's code, I realized

that I had accidentally assigned the parameter “numberOfProfessors” to “this.numberOfYears” which is the variable being used for the rest of the class’s methods. After replacing this line with “this.numberOfYears = numberOfYears,” the program was fixed and started functioning as expected.

Because all of the methods involved a very similar nested for loop to retrieve data from the array of salaries, I was able to quickly define the methods. Returning the total amount of money earned by all professors was very easy after already defining a method for returning the average salary of all the professors. All I needed to do was return the total without dividing by the number of professors. Because of the work I did earlier on, implementing these methods was a breeze! My understanding of things like instance variables, nested loops, arrays, constructors, and general code organization allowed me to think ahead and structure this project in a way that would allow me to implement new methods without having to worry about breaking previous functionality. The lectures and assignments provided this semester not only allowed me to understand the “how” related to implementing good java code, it also allowed me to understand the “why,” which I attribute to my ability to think ahead and better organize my code for both initial and future development!