In this app, I am creating an application that will determine gross pay for a list of employees.

To accomplish this, I opt to use the Java ArrayList library, a float[] array, the Scanner library to take user input, and a try{}catch to catch errors.

First, I begin by initializing most of the variables I need, which includes employeeId, hoursWorked, hourlyRate, grossPay, etc. I also initialize an empty ArrayList which contains Float[] arrays, which will contain the employee data. Additionally, I initialize a Boolean isValid to false, to check user input.

I begin a while loop to take user input, asking for comma separated lists and storing them in temporary string variables. After each user input is taken, I convert the strings containing comma separated lists into String[] arrays, by using the String.split() method, with a “,” for regex.

After this step is completed, I initiate the try, where iterate through the employeeId array and create a new Float[] array using the index variable to adjacently iterate through the employeeHourlyRates array, and the employeeHoursWorked array. I cast each element of those arrays as floats and create a temporary Float[] array variable during that execution of the for loop. Then, I add this float[] variable to the employees ArrayList, after which the execution repeats until the loop reaches the end of employeeIdArray. If the userinput is incorrect, the catch will catch the exception and tell the user to correctly input the data. Unfortunately, this does not catch the pitfall that they may enter data in the incorrect order.

Once the for loop completes and no exceptions are encountered, isValid is set to true and we exit the while loop.

Now, I begin to get ready to display the output to the user. I use the System.out.printf method to print to the console with strict spacing, for better legibility. Then, I initiate a for loop that iterates through the employees ArrayList, extracts the values for each employee from that element of the ArrayList, performs the calculation required for grossPay, and outputs to the console in a table-like format.