This assignment was a little more involved than previous modules. I came across a lot of issues with code provided in the textbook being deprecated which required me to find an equivalent or an alternative way of achieving the goals. Other than that, the assignment was not that difficult.

I started out by creating two functions, one to get the current datetime and the other to count the number of rows inside a file. Starting at the top of the file I addressed each of the TODO’s. I edited the encrypt\_decrypt function, adding in the XOR encryption.

The next function I modified was read\_file, this function was heavily modified to reach the desired functionality. I start out by setting up a try/catch and then declare a variable for a file name and assigned it the value passed to the function. This was followed by opening the file, created the rows variable and assigned it the value returned by the rowCount function. Next, I perform a check to make sure the file was opened, it if was I then check if there is data in the file, if there is not, throw an error, otherwise continue. Using a while loop I get the data from the file line by line using the getline function and assigned the data to a variable line. The variable line is then added to file\_text.

It is at this point where I deviate a little, given that I am OCD bad I could not stand the fact my name was being shown twice on the decrypted file. I understand why it was doing it but didn’t like it. So, with that said, the read\_file I changed the name of the string variable that was originally inside this function to assignment which shows this assignment. After each line in the file has been read and added to the file\_text variable a pos variable is created in which I use the find method to locate the end of the line with my name in it, using the insert method I insert the assignment into the file\_text after my name. From there I close the file and return file\_text.

The next function I had to modify was save\_data\_file, the first thing I do is declare the timestamp variable and call the getDateTime function to get the current timestamp. I use a try/catch here as well to intercept any errors encountered. Much like I did in the read\_file function I create a filename variable and assign it the value passed to the function. Next is checking to make sure the file opened and then I call the clear method to make sure the file we are going to be saving data into is empty. I understand in a production environment that would be a major issue but it makes testing very easy for small scale. Using the parameters passed to the function I then send each piece of data to the file and finally I close the file.

Going back to the read\_file function where my OCD kicked in, the only thing I changed in the main function was after the value of student\_name is assigned I create a pos variable to locate the position of the last character on row one of the sources string variables. Using that position I use the erase method to completely remove my name from the string. The reason I did this is because in the save\_data\_file function we add the student’s name to the file. That produced the screenshot shown below.

|  |
| --- |
| Text  Description automatically generatedConsole Output |
| Text  Description automatically generatedDecrypted File |