Project 2

I think some of the notable obstacles I overcame was mainly regarding the folder layout and file locations. Often times, I am still trying to get use to the system in which we use g++ with Linux and remote computer access in order to perform the debugging or checks. It is confusing how the solution has the same name as the project folders at time, so it makes navigation a bit more clustered when trying to locate the main cpp files needed to submit for the projects. Although I wasted a lot of time trying to set up my own Linux access and putty server from downloading Xlaunch, Xming, and Cisco AnyConnect, I realized the optimal approach was to find access to the remote server and computers in order to check my program. Ironically, the compiler process was the area that I struggled the most on. The only obstacle I faced regarding the source code was simply a mistake between the forward and back slash for the cin.ignore(10000, ‘\n’); but instead, I used a forward slash (‘/n’).

List of test data:

Name (Jason Bourne, Conner Henry, Matt, William, Jo Jo)

Taxable income (64000, 500000, 20, -40, 0)

Occupation (engineer, teacher, , scientist, doctor)

Number of children (4, 1000, 4, 0, 1)

Test case 1: The program takes in the occupation and children to modify the tax rate to be lower compared to someone who isn’t an engineer or scientist or does not have any children.

Test case 2: The program should know to not intake the number of children given the exceeding amount of taxable income from the individual.

Test case 3: an empty occupation input should trigger the response of “You must enter an occupation” once all the data is inputted. Similar results are outputted accordingly if there is an empty input for name, income, and children; however, empty inputs where there should be a numeric value would create an error in the system’s chronological functions.

Test case 4: After all input is submitted, a response of a negative value for taxable income would lead to the output of “The taxable income must be nonnegative.” (Similarly, for children)

Test case 5: The program would compute a negative result from the arthimetic operations given for the system to carry out leading to the result of a negative sum needed to pay. The system would resolve the negative tax results by defaulting the value to $0. Interestingly enough, this condition holds assuming the integer inputted for children is less than or equal to numbers