Eduardo Ramirez Torres

UID: 205317677

CS31, Discussion 1D

**Report for Project 2**

1. The project specs were easy enough to understand and follow. However, when it came to making the if statements, I ran into an obstacle. I struggled a little bit on how to structure the statements such that I can check each type of situation in a correct manner. Eventually, I decided to first check if the student paid over $40k; once that was checked, I would then have to check if they paid over $250k; finally, for this instance, I could calculate the fine for the student if they are a legitimate athlete or not. If the student did not pay over $250k, then it would fall into the else statement that just calculated the fine for over $40k and under $250k, given if the student was a legitimate athlete or not. If the student paid under $40k, regardless of their legitimacy as an athlete (as specified in the project), there would be a single calculation for that instance. After solving that issue, everything else in the project went swimmingly.
2. In order to make sure that my code worked, I used some test data:
   1. Blank student name (“”, 75, n): this makes sure that the code can detect the empty string and display the message for that instance.
   2. Negative number (Name McNameface, -75, n): similar to the previous situation, but for the amount paid.
   3. Invalid y/n input (Some Name, 75, w): again, like previously, but for the y/n input.
   4. A ridiculously high number (Another Name, 1000000000000000000, y): tests the limits of the program.
   5. A ridiculously long name (Supercalifragilisticexpialidocious [floccinaucinihilipilification, 75, n): tests limits of getline.](https://www.lexico.com/en/definition/floccinaucinihilipilification)
   6. Test cases for amount paid (e.g.: 20, 67, 260): checks to see if calculations are correct for the program